

## Research Article

# Multiobjective Optimization Method Based on Adaptive Parameter Harmony Search Algorithm

P. Sabarinath,<sup>1</sup> M. R. Thansekhar,<sup>1</sup> and R. Saravanan<sup>2</sup>

<sup>1</sup>K.L.N College of Engineering, Pottapalayam 630611, India

<sup>2</sup>Sri Krishna College of Technology, Coimbatore 641 042, India

Correspondence should be addressed to P. Sabarinath; [jananisabari@gmail.com](mailto:jananisabari@gmail.com)

Received 17 October 2014; Revised 8 January 2015; Accepted 8 January 2015

Academic Editor: Zong Woo Geem

Copyright © 2015 P. Sabarinath et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The present trend in industries is to improve the techniques currently used in design and manufacture of products in order to meet the challenges of the competitive market. The crucial task nowadays is to find the optimal design and machining parameters so as to minimize the production costs. Design optimization involves more numbers of design variables with multiple and conflicting objectives, subjected to complex nonlinear constraints. The complexity of optimal design of machine elements creates the requirement for increasingly effective algorithms. Solving a nonlinear multiobjective optimization problem requires significant computing effort. From the literature it is evident that metaheuristic algorithms are performing better in dealing with multiobjective optimization. In this paper, we extend the recently developed parameter adaptive harmony search algorithm to solve multiobjective design optimization problems using the weighted sum approach. To determine the best weightage set for this analysis, a performance index based on least average error is used to determine the index of each weightage set. The proposed approach is applied to solve a biobjective design optimization of disc brake problem and a newly formulated biobjective design optimization of helical spring problem. The results reveal that the proposed approach is performing better than other algorithms.

## 1. Introduction

Engineering design optimization has recently received lot of attention from designers so as to produce better designs. It saves time, cost, and energy involved. Engineering design optimization often deals with many design objectives under nonlinear and complex constraints. Also the design problem is subjected to constraints limited by cost, weight, and material properties like strength, design specifications, and availability of resources. The design objectives are often conflicting in nature and hence finding the true Pareto optimal front is difficult. Conventionally, numerous mathematical methods such as linear, nonlinear, dynamic, and geometric programming have been developed in the past to solve engineering design optimization problems. But these methods have many drawbacks and no single method is found suitable for solving all types of engineering design optimization problems.

Even for a single objective design optimization problem involving large number of nonlinear design variables,

arriving at a global best solution is not an easy task. The shortcoming of mathematical methods gives way for metaheuristics algorithm to solve engineering design optimization problems. Metaheuristics algorithms have proven to be powerful in solving this kind of optimization since they combine rules and randomness to mimic natural phenomena such as biological systems (Genetic Algorithm), animal behavior (ant algorithm, tabu search), and physical annealing process (Simulated Annealing). GA is a popular metaheuristics algorithm which has been broadly applied to solve various design optimization problems and proven to be successful in finding the global best than traditional methods. These include the design optimization of machine elements such as gears [1], gearbox [2], journal bearing [3], magnetic thrust bearing [4], rolling element bearing [5], and automotive wheel bearing unit [6]. Swarm intelligence is another important concept in many recent metaheuristics algorithms such as particle swarm optimization, firefly algorithm, artificial bee colony algorithm, and cuckoo search.

Particle swarm optimization mimics the group behavior of birds and fish schools [7]. Sabarinath et al. [8] have applied PSO algorithm for solving optimal design of belt pulley system. Yang [9] developed firefly algorithm that imitates the flashing behavior of tropic firefly swarms. Yang and Deb [10] also proposed cuckoo search algorithm based on the behavior of cuckoo birds. Recently Geem et al. [11] proposed a new harmony search (HS) that draws its motivation not from a biological or physical process like most other metaheuristic optimization algorithms, but from an arty one, that is, the improvisation process of musicians in the hunt for a magnificent harmony. Geem et al. [11] explained the similarity between musical performance and optimization that explains the strength and language of HS. The attempt to find the harmony in music is similar to finding the optimality in an optimization process and the musician's improvisations are similar to local and global search schemes in optimization techniques. HS algorithm uses random search based on harmony memory considering rate (HMCR) and the pitch adjusting rate (PAR) instead of gradient search. Even though the basic algorithm is successful in solving variety of problems still there is a growing diversity of modified HS algorithms that try to find improved performance. Recently Kumar et al. [12] had proposed a new version of HS algorithm, namely, parameter adaptive harmony search (PAHS) to solve standard benchmark functions in single objective optimization successfully. This PAHS algorithm is extended to solve multiobjective design optimization problems by weighted sum approach. In this weighted sum PAHS approach, a performance index based on least average error [13] is used to evaluate the performance of each weightage set.

## 2. Literature Review

*2.1. Previous Work on Disc Brake Optimization.* The proposed multiobjective disc brake optimization problem was introduced by Osyczka and Kundu [21]. The authors used the modified distance method in genetic algorithm to solve the disc brake problem and compared their results with that of a plain stochastic method. Ray and Liew [22] used a swarm metaphor approach in which a new optimization algorithm based on behavioural concepts similar to real swarm was proposed to solve the same problem. Yıldız et al. [23] used hybrid robust genetic algorithm combining Taguchi's method and genetic algorithm. The combination of genetic algorithm with robust parameter design through a smaller population of individuals resulted in a solution that lead to better parameter values for design optimization problems. L16 orthogonal arrays were considered to design the experiments for this problem. The optimal levels of the design parameters were found using ANOVA with respect to the effects of parameters on the objectives and constraints. Yıldız [24] used hybrid method combining immune algorithm with a hill climbing local search algorithm for solving complex real-world optimization problems. The results of the proposed hybrid approach for multiobjective disc brake problem were compared with the previous solutions reported in literature.

Yang and Deb [25] used multiobjective cuckoo search (MOCS) algorithm for solving this disc brake problem by considering two objective functions. The authors extended the original cuckoo search for single objective optimization by Yang and Deb for multiobjective optimization by modifying the first and third rule of three idealized rules of original cuckoo search. The same author [26] used multiobjective firefly algorithm (MOFA) for solving this disc brake problem. By extending the basic ideas of FA, Yang et al. developed multiobjective firefly algorithm (MOFA). Yang et al. [27] successfully extended a flower algorithm for single objective optimization to solve multiobjective design problems. The author solved the biobjective disc brake problem using multiobjective flower pollination algorithm (MOFPA). Reynoso-Meza et al. [28] used the evaluation of design concepts and the analysis of multiple Pareto fronts in multicriteria decision-making using level diagrams. They addressed multiobjective design optimization problem of disc brake by considering the friction surfaces as 4 and 6 to obtain Pareto fronts.

*2.2. Previous Work on Helical Compression Spring Optimization.* In single objective spring design optimization problem, two different cases were proposed with objective function of volume minimization as case I and weight minimization as case II. This single objective design problem had been solved using many optimization algorithms as two different cases. In this section we will look into a few reviews on case I of a compression spring design problem since the proposed multiobjective helical compression spring optimization problem is the conversion of well-known standard benchmark problem of single objective design optimization, that is, volume minimization of helical compression spring (case I). For both the cases, the design variables are common. The three design variables are the wire diameter  $d = x_1$ , the mean coil diameter  $D = x_2$ , and the number of active coils  $N = x_3$ . But the data type of design variables, objective function, and constraints of these two cases are different. Sandgren [15] used integer programming to solve this problem. Chen and Tsao [16] used simple genetic algorithm to minimize the volume of the helical compression spring. Wu and Chow [17] used metagenetic parameter in GA to solve the same problem. Another improved version of GA called geneAS was used by Deb and Goyal [29] to address this problem. The same problem was solved by discrete version of PSO by Kennedy and Eberhart [20] and by using differential evolution algorithm by Lampinen and Zelinka [19]. Guo et al. [18] used swarm intelligence to optimize the design of helical compression spring. Datta and Figueira [30] used real-integer-discrete-coded PSO for solving this case I problem. He et al. [31] used an improved version of PSO to solve this case I spring problem. Deb et al. [32] proposed NSGA II algorithm for solving some biobjective design optimization of mechanical components. He used case I spring problem and converted it to a biobjective problem by adding one more objective of minimizing stress induced in the spring. In this work, we have converted the case I spring problem into a biobjective problem by adding one more objective of maximizing the strain energy stored in the spring. In all works

related with case I spring problem, the previous researchers used FPS unit system. But in this work, all data are converted into SI units and the results of this proposed multiobjective spring problem are presented in SI units.

The main objectives of this work are (i) multiobjective design optimization of the disc brake and compression spring using the weighted sum approach of parameter adaptive harmony search algorithm and (ii) to demonstrate the effectiveness of this proposed algorithm for multiobjective optimization of machine components. So far, the weighted sum approach of parameter adaptive harmony search algorithm has not been tried or experimented for the multiobjective optimization of machine components. In this paper, ability of the algorithm is demonstrated using a performance index based on least average error. The optimization results obtained by using this proposed algorithm are compared with those obtained by previous researchers using other methods.

**2.3. Previous Works on HS Algorithm.** HS algorithm is widely used in solving various optimization problems such as pipe network design [33], design of coffer dam drainage pipes [34], water distribution networks [35], satellite heat pipe design [36], design of steel frame [37], power economic load dispatch [38], economic power dispatch [39], power flow problem [40], vehicle routing [41], orienteering [42], robot application [43], and data clustering [44]. Mahdavi et al. [45] proposed improved HS algorithm by changing the parameters of HS algorithm dynamically. HS algorithm has been implemented by many researchers for the past few years to solve optimization problems in many fields of engineering, science, and technology. To increase the performance of HS algorithm, several improvements were done periodically in the past. From the literature, it is observed that mainly two improvements had been considered. First improvement is in terms of tuning the parameters of HS algorithm and the other one is in terms of hybridizing the components of HS algorithm with other metaheuristic algorithms. In this work, we are concentrating only with the first improvement.

Mahdavi et al. [45] first attempted an improvement in HS algorithm by proposing a new variant of HS known as improved HS (IHS) algorithm. They changed the variants such as pitch adjusting rate (PAR) and bandwidth (BW) dynamically with generations. Linear increase in PAR and exponential decrease in BW were allowed between prespecified minimum and maximum range.

Kong et al. [46] proposed an adaptive HS (AHS) algorithm to adjust PAR and BW. In this approach, PAR was changed dynamically in response to objective function values while BW was tuned for each variable. Omran and Mahdavi [47] proposed a global best HS algorithm inspired by PSO algorithm to improve the performance of HS algorithm. The difficulty in finding the lower and upper bounds of BW was solved by using global best particle which is the fittest particle in the swarm in terms of objective function values than other particles. A self-adaptive mechanism for selecting BW was proposed by Das et al. [48], known as explorative HS algorithm. BW was recomputed for each iteration while the other parameters were kept fixed. The other improvements in

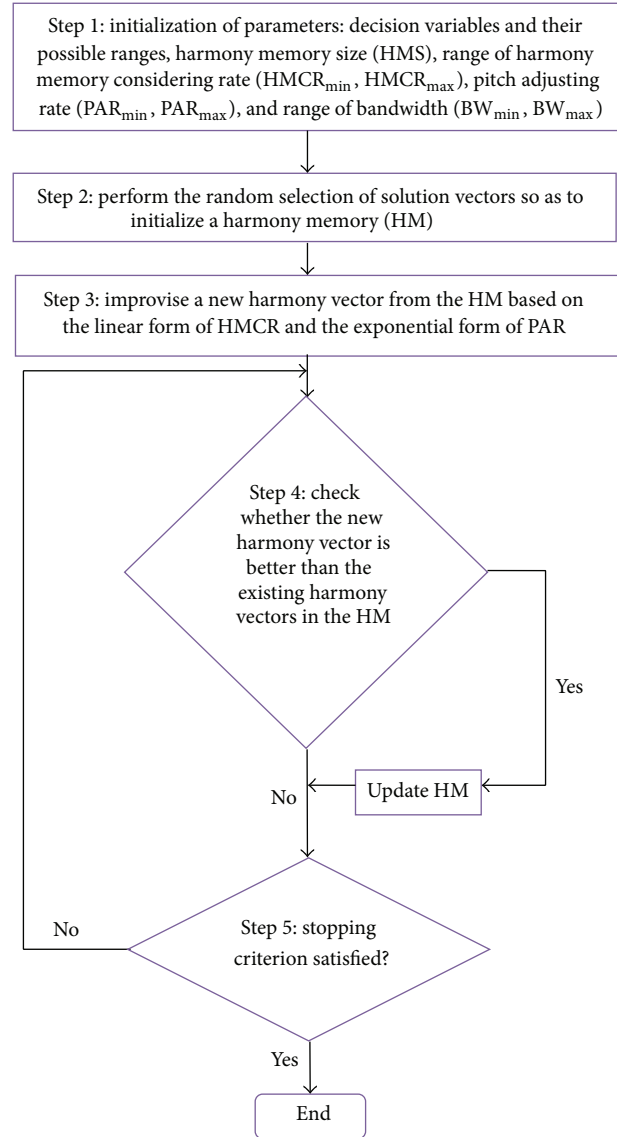


FIGURE 1: Flow chart of the proposed PAHS optimization approach.

HS variants were reported in the literature [14, 41, 44, 48–55]. In all the above cases, only PAR and BW values were changed whereas HMCR was kept fixed. This fixed value of HMCR is the key factor which prevents to get global optimal solution. In order to overcome this difficulty, a parameter adaptive harmony search (PAHS) algorithm was recently proposed by Kumar et al. [12]. A detailed description of the proposed parameter adaptive harmony search (PAHS) algorithm can be seen from [12]. However, for the sake of easiness a brief introduction and a step by step computational procedure for implementing this algorithm are given in the following section. A flow chart showing the Steps 1 to 5 of the proposed PAHS approach is given in Figure 1.

### 3. Parameter Adaptive Harmony Search (PAHS) Algorithm

The harmony search algorithm (HS) is one of the most recent metaheuristic optimization algorithms, analogous to music

improvisation process where the musicians improvise the pitches of their instruments to obtain better harmony. Mahdavi et al. [45] further improved HS in the form of improved harmony search algorithm (IHS) wherein the parameters are changed dynamically. Harmony memory consideration rate (HMCR) and pitch adjustment rate (PAR) are the two important parameters that largely affect the performance of HS algorithm. While searching for global best solution, we make use of HMCR, which in turns requires proper tuning, failing which the solution may stuck with local optima. PAR plays a vital role as far as the local search is concerned. During improvisation process, the fixed value of these parameters may lead to a delay in convergence because of the absence of proper balance of global and local search capabilities. The only solution for this issue may be obtained by changing both HMCR and PAR dynamically. The above issue is addressed in the newly proposed algorithm named as parameter adaptive harmony search algorithm (PAHS) by Kumar et al. [12]. In PAHS, a dynamic change in PAR and HMCR values were proposed consequently by modifying improvisation step of IHS. The value of HMCR is initially kept small in order to explore the entire search space to a larger extent and later on the value of HMCR is increased to confine the search space HM (local search) only. Similarly, the value of PAR is initially kept high in order to modify the solutions either stored in HM or from feasible range. Finally PAR value is reduced to obtain better solutions. Based on the modifications stated above, (1) in linear form for HMCR and (2) in exponential form for PAR had been proposed as given below:

$$\text{HMCR}(\text{gn}) = \text{HMCR}_{\min} + \frac{(\text{HMCR}_{\max} - \text{HMCR}_{\min})}{\text{NI}} \times \text{gn}, \quad (1)$$

$$\begin{aligned} \text{PAR}(\text{gn}) \\ = \text{PAR}_{\max} \cdot e \left( \frac{\ln(\text{PAR}_{\min}/\text{PAR}_{\max})}{\text{NI}} \times \text{gn} \right), \end{aligned} \quad (2)$$

where  $\text{HMCR}(\text{gn})$  is the harmony memory consideration rate for generation  $\text{gn}$  and  $\text{HMCR}_{\min}$  and  $\text{HMCR}_{\max}$  are the minimum and maximum values of harmony memory consideration rate, respectively. Similarly  $\text{PAR}_{\min}$  and  $\text{PAR}_{\max}$  are the minimum and maximum values of pitch adjustment rate, respectively. NI represents the maximum number of improvisations and  $\text{gn}$  represents generation number. The bandwidth is given as

$$\text{BW}(\text{gn}) = \text{BW}_{\max} \cdot e \left( \frac{\ln(\text{BW}_{\min}/\text{BW}_{\max})}{\text{NI}} \times \text{gn} \right), \quad (3)$$

where  $\text{BW}(\text{gn})$  is the bandwidth for generation  $\text{gn}$  and  $\text{BW}_{\min}$  and  $\text{BW}_{\max}$  are the minimum and maximum values of bandwidth, respectively.

They have considered four different cases wherein besides the linear change, two parameters have been allowed exponential change also. PAHS's variants were applied on fifteen standard benchmark test functions. They have compared the results of PAHS with the other versions of HS as proposed by Geem et al. [11], Mahdavi et al. [45], and Kong et al.

[46]. The authors also applied this PAHS to data clustering problem. From their results it is evident that the PAHS-3, that is, linear change in HMCR and exponential change in PAR, has outperformed the results of HS, IHS, AHS, and other variants of HS. PAHS-3 also provided better results in both dimensional and noisy environments. Hence, in this work, PAHS-3 is used to solve multiobjective design optimization problems using weighted sum approach.

### 3.1. Computational Procedure of PAHS

*Step 1* (initialization). In this step, the optimization problem and the algorithmic parameters are initialized. In this paper, the minimization of objective function is considered for the two design optimization scenarios.

The optimization problem is stated as minimize  $f(x)$  such that  $x = (x_1, x_2, \dots, x_n)$  is the decision variable set.  $n$  is the number of decision variables.  $\text{UB}_i$  and  $\text{LB}_i$  are the upper and lower bounds for decision variable  $x_i$ , respectively. The parameters of the PAHS algorithm are harmony memory size (HMS), range for harmony memory consideration rate ( $\text{HMCR}_{\min}$ ,  $\text{HMCR}_{\max}$ ), range for pitch adjustment rate ( $\text{PAR}_{\min}$ ,  $\text{PAR}_{\max}$ ), range for distance bandwidth ( $\text{BW}_{\min}$ ,  $\text{BW}_{\max}$ ), and number of improvisation (NI).

*Step 2* (initialize harmony memory (HM)). Solution vectors are randomly generated for the size of HM, that is, (HMS) in order to fill the harmony memory. Next sorting of harmony memory is done based on the values of the objective function  $f(x)$ .

*Step 3* (generation of new harmony). Using the procedure given below, a new harmony vector  $x^1 = (x^1_1, x^1_2, \dots, x^1_n)$  is generated

```

For each  $i \in [1, n]$  do
   $\text{HMCR} = \text{HMCR}(\text{gn})$  /* Proposed equation (1) */
   $\text{PAR} = \text{PAR}(\text{gn})$  /* Proposed equation (2) */
   $\text{BW} = \text{BW}(\text{gn})$  /* (3) */
  if  $U(0, 1) \leq \text{HMCR}$  then /* memory consideration */
  begin
     $x'_i = x_i^l$  where  $l \sim U(1, \dots, \text{HMS})$ 
  if  $U(0, 1) \leq \text{PAR}$  then /* pitch adjustment */
  begin
     $x'_i = x'_i \pm \text{BW} \times \text{Rand}$ ,  $\text{Rand} \sim U(0, 1)$ 
  endif
  else /* random selection */
     $x'_i = \text{LB}_i + (\text{UB}_i - \text{LB}_i) \times \text{Rand}$ 
  endif
  done

```

*Step 4* (update harmony memory). Evaluate the newly generated harmony vector  $x^1 = (x^1_1, x^1_2, \dots, x^1_N)$  in terms of objective function value. It is compared with that of the initial

harmony memory. The better harmony vector is added in the harmony memory and the worst harmony vector is dropped from the harmony memory. This step is the deciding stage of the algorithm whether a new harmony vector is to be included in the harmony memory or not.

*Step 5* (check for the termination criterion). If the maximum number of improvisation step (termination criterion) is reached, computation is terminated and the best harmony vector is returned by the algorithm. Otherwise, repeat Steps 3 and 4 till the best harmony vector is obtained.

For solving the two design optimization problems of machine elements (disc brake and compression coil spring) explained in Section 4 using weighted sum approach of PAHS, the harmony memory size (HMS) is set to 20, and the maximum generations (set as stopping condition) are set to 200 generations. The harmony memory consideration rate (HMCR) is varied in the range of 0.7 to 0.99 while the pitch adjustment rate (PAR) is varied in the range of 0.01 to 0.99. The bandwidth (BW) is varied in the range of 0.001 to  $1/20(UB-LB)$ . Static penalty method is applied for handling the constraints. A total of 4000 fitness function evaluations were made with this optimization approach in each run. To study the performance of proposed parameter adaptive harmony search algorithm, constrained optimization of input parameters to minimize the objective functions was solved and the best results obtained through the mentioned optimization approach in 40 trials were compared with those reported in the literature.

*3.2. Weight Based Multiobjective Optimization Approach.* The easiest way of implementing the multiobjective optimization is done through weighted sum method. In this approach, the parameter adaptive harmony search algorithm is modified to facilitate multiobjective design optimization. The implementation of weighted sum approach by assigning suitable weight values to represent the preference in choice is explained by Marler and Arora in multiobjective optimization (MOO) [56]. The implementation of weighted sum approach to artificial bee colony algorithm is addressed in Hemamalini and Simon [57] where the multiobjective optimization is treated as composite objective function. The composite objective function is expressed as follows:

$$f(x) = w_1 f_1(x) + w_2 f_2(x), \quad (4)$$

where  $w_1$  and  $w_2$  are the assigned weights. The relationship between the variables  $w_1$  and  $w_2$  is given by  $w_2 = 1 - w_1$  where  $w_1$  is chosen in the range of [0-1].

Min max normalization is performed for the objective functions so as to assess the fitness of the composite objective function. The normalization is done for scaling the objective functions within a specified range. The following normalization formula is used [58]:

$$fit'_i = \frac{fit_i - \min(fit_{overall}) \times \partial}{\max(fit_{overall}) - \min(fit_{overall}) \times \partial}, \quad (5)$$

where  $fit_i$  represents the fitness to be normalized and  $fit_{overall}$  represents the overall fitness.  $\partial$  is taken as 0.999 in order to avoid zeroes during normalization process.

*3.3. Selection of Weightage Value and Performance Index.* An analysis is performed by changing the weight values  $w_1$  from 1 to 0 with a step size of 0.1 so as to assign the proper weightage values in (4). It is done to see the importance of each weightage set ( $w_{set}$ ) towards the objective function. A performance index based on the least average error (LAE) formulated by Naidu et al. [13] is used to evaluate the performance of each weightage set. The LAE is evaluated by

$$LAE = \left( \left( \frac{F_{1,w_{set},i} - \min(F_{1,overall})}{\min(F_{1,overall})} \right) + \left( \frac{F_{2,w_{set},i} - \min(F_{2,overall})}{\min(F_{2,overall})} \right) \right) \times (2)^{-1}, \quad (6)$$

where  $F_{1,w_{set},i}$  represents fitness 1 value and  $F_{2,w_{set},i}$  represents fitness 2 value at weightage set  $I$  (where  $I = 1, 2, \dots, 11$ ).  $F_{1,overall}$  and  $F_{2,overall}$  represent the overall fitness 1 and fitness 2 values. The performance index based on LAE ( $\eta_{LAE}$ ) is expressed as

$$\eta_{LAE} = \frac{1}{LAE}. \quad (7)$$

## 4. Problem Formulation

*4.1. Multiobjective Disc Brake Optimization Problem.* The multiobjective disc brake optimization problem was solved by Osyczka and Kundu [21] using plain stochastic method and genetic algorithms for optimization of disc brake problem. They have shown that genetic algorithm was giving better results compared with that of plain stochastic method. The objectives of the problem are to minimize the mass of the brake and to minimize the stopping time. The disc brake optimization model has four variables (as shown in Figure 2) that are

- (1)  $R_i$ , inner radius of the discs, in mm =  $x_1$ ,
- (2)  $R_o$ , outer radius of the discs, in mm =  $x_2$ ,
- (3)  $F$ , engaging force, in N =  $x_3$ ,
- (4)  $n$ , number of the friction surfaces (integer) =  $x_4$ .

The objective functions and constraints of the disc brake design optimization model provided by Osyczka and Kundu [21] are defined as follows.

### Objective Functions

Mass of the brake:

$$f_1(x) = 4.9 \times 10^{-5} (x_2^2 - x_1^2) (x_4 - 1), \quad \text{in kg.} \quad (8)$$

Stopping time:

$$f_2(x) = \frac{9.82 \times 10^6 (x_2^2 - x_1^2)}{x_3 x_4 (x_2^3 - x_1^3)}, \quad \text{in s.} \quad (9)$$

The constraints are as follows.

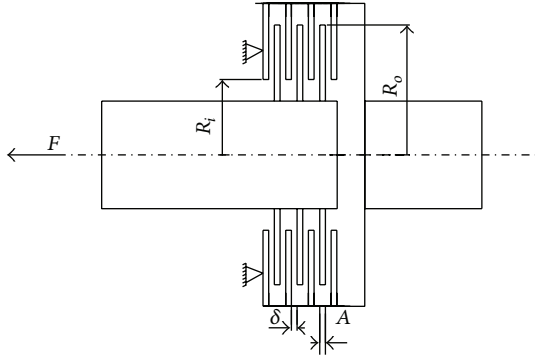


FIGURE 2: The disc brake design problem.

(1) Side constraints  $55 \leq x_1 \leq 80$ ,  $75 \leq x_2 \leq 110$ ,  $1000 \leq x_3 \leq 3000$ ,  $2 \leq x_4 \leq 20$ .

(2) Geometric constraints

(i) Minimum distance between radii:

$$g_1(x) = (x_2 - x_1) - 20 \geq 0. \quad (10)$$

(ii) Maximum length of the brake:

$$g_2(x) = 30 - 2.5(x_4 + 1) \geq 0. \quad (11)$$

(3) Behaviour constraints

(i) Pressure constraint

$$g_3(x) = 0.4 - \frac{x_3}{3.14(x_2^2 - x_1^2)} \geq 0. \quad (12)$$

(ii) Temperature constraint

$$g_4(x) = 1 - \frac{2.22 \times 10^{-3} x_3 (x_2^3 - x_1^3)}{(x_2^2 - x_1^2)^2} \geq 0. \quad (13)$$

(iii) Generated torque constraint

$$g_5(x) = \frac{2.66 \times 10^{-2} x_3 x_4 (x_2^3 - x_1^3)}{(x_2^2 - x_1^2)} - 900 \geq 0. \quad (14)$$

**4.2. Compression Coil Spring Design: Case I.** This problem involves the design of a coil spring, which is to support a constant axial compressive load. It is required to minimize the wire volume of the spring so that it can support a given load without failure. At the same time, it is desired to store maximum strain energy possible. So these two objectives are conflicting in nature. As shown in Figure 3, there are three design variables: the real-valued outside diameter of the spring ( $D$ ), the integer-valued number of spring coils ( $N$ ), and the discrete-valued spring wire diameter ( $d$ ).

The objective, in this problem, is to find the optimal combination of the above three design parameters that provides minimum weight and maximum storage of energy subjected to the constraints.

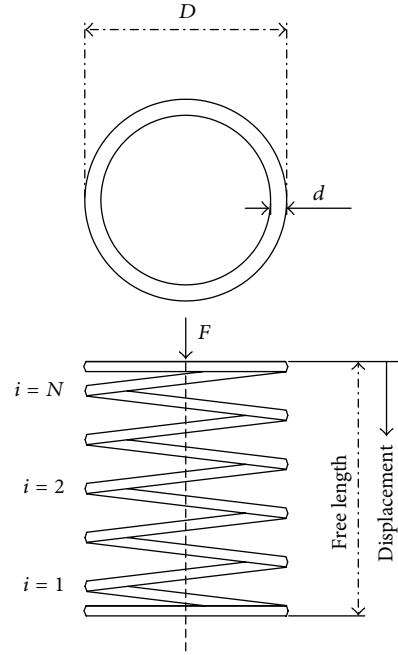


FIGURE 3: The compression coil spring design problem.

Accordingly, the biobjective optimization problem is formulated as below:

$$\text{Determine } \mathbf{x} = (D, N, d). \quad (15)$$

To minimize the weight of the spring, that is,

$$F_1(x) = \frac{\pi^2}{4} D d^2 (N + 2), \quad (16)$$

and to maximize the strain energy stored, that is,

$$F_2(x) = \frac{1}{2} \times P \times \partial,$$

$$F_2(x) = \frac{1}{2} \times P \times \frac{8PD^3N}{Gd^4},$$

$$F_2(x) = \frac{4F_{\max}^2 D^3 N}{Gd^4},$$

$$\text{Subjected to } g_1(x) = \frac{8CKF_{\max}}{\pi d^2} - S \leq 0,$$

$$g_2(x) = l - l_{\max} \leq 0,$$

$$g_3(x) = d_{\min} - d \leq 0,$$

$$g_4(x) = (D + d) - D_{\max} \leq 0,$$

$$g_5(x) = 3.0 - C \leq 0,$$

$$g_6(x) = \delta_p - \delta_{pm} \leq 0,$$

$$g_7(x) = \delta_p + \frac{F_{\max} - F_p}{k}$$

$$+ 1.05 (N + 2) d - l \leq 0,$$

$$g_8(x) = \delta_w - \frac{F_{\max} - F_p}{k} \leq 0, \quad (17)$$

where

$$C = \frac{D}{d}, \quad K = \frac{4C - 1}{4C - 4} + \frac{0.615}{C},$$

$$k = \frac{Gd}{8NC^3}, \quad \delta_p = \frac{F_p}{k}, \quad (18)$$

$$l = \frac{F_{\max}}{k} + 1.05 (N + 2) d.$$

The supplied numerical data for the problem are given in SI units in Table 1 along with the allowable discrete values of the spring wire diameter ( $d$ ) in Table 2.

The present work aims to minimize the volume of the spring and maximize the energy stored. Since the objectives are conflicting in nature, modification of second objective, that is, strain energy (SE), is done to get it converted for minimization.

The objective functions are given below:

$$\text{Objective 1} = \text{Volume of the helical spring,}$$

$$\text{Objective 2} = \frac{1}{\text{S.E.}} \quad (19)$$

## 5. Results and Discussion

The multiobjective PAHS algorithm has been applied for constrained design optimization of input parameters of disc brake and helical compression spring problems. The optimization process has been implemented using Matlab 2009 to run on a PC compatible with Pentium IV, a 3.2 GHz processor, and 2 GB of RAM (Random Access Memory).

**5.1. Disc Brake Problem.** The weightage set is varied in the range of 0 to 1 with a step size of 0.1. It is evident from the results that single objective optimization is achieved, when the weights are defined at their extreme values. The optimal design parameters of disc brake at different weights are shown in Table 3. When  $w_1 = 1$  and  $w_2 = 0$ , minimizing the mass of the brake is the effective objective function and it can be observed that the mass of the brake is minimum but the stopping time value is large. In the other extreme, when  $w_1 = 0$  and  $w_2 = 1$ , minimizing the stopping time of the brake is the effective objective function in which the stopping time is minimum but the mass of the brake is large. These results are in accordance with the literature discussed above. By varying the weightage set with a step size of 0.1, we can generate 11 possible combinations. In order to find the best combination of weightage set, a performance index based on least average error (LAE) is used as explained in Section 3.2. The purpose of this performance index is to find the least possible error for both objective functions. The first part of (5) is to find the lowest error ratio for the first objective function, that is,

TABLE 1: Numerical data for spring problem.

Parameter name	Symbol	Numerical value
Maximum working load	$F_{\max}$	4448.22 N
Maximum allowable shear stress	$S$	1303.109 MPa
Maximum free length	$l_{\max}$	355.6 mm
Minimum wire diameter	$d_{\min}$	5.08 mm
Maximum outside spring diameter	$D_{\max}$	76.2 mm
Preload compression force	$F_p$	1334.47 N
Maximum allowable deflection under preload	$\delta_{pm}$	152.4 mm
Deflection from preload position to maximum load position	$\delta_w$	31.75 mm
Shear modulus of the material	$G$	79289.32 MPa

TABLE 2: Allowable wire diameters (discrete values of  $d$ ).

Allowable wire diameters (mm)						
0.2286	0.2413	0.26416	0.29972	0.32512	0.33528	0.3556
0.381	0.41148	0.43942	0.4572	0.508	0.5842	0.635
0.7112	0.8128	0.889	1.0414	1.1938	1.3716	1.6002
1.8288	2.032	2.3368	2.667	3.048	3.429	3.7592
4.1148	4.4958	4.8768	5.2578	5.715	6.1976	6.6802
7.1882	7.7978	8.4074	9.1948	10.0076	11.1125	12.7

mass of the brake. This is done by subtracting the minimum error value ( $\min F_{1,\text{overall}}$ ). Every objective function value of mass of the brake in Table 3 will be subtracted from this particular value and then the resulting value will be divided by the minimum error value ( $\min F_{1,\text{overall}}$ ) to obtain the ratio. A negative answer would not appear through this method since we are subtracting the least value. The same principle is applied to the second objective function, that is, stopping time of the brake. The results for both the calculations are then averaged to obtain a single index value. Based on the highest  $\eta_{\text{LAE}}$  value observed in Table 3, the weightage set of  $w_1 = 0.9$  and  $w_2 = 0.1$  is chosen as the best combination. Figure 4 shows the plot of performance index based on LAE ( $\eta_{\text{LAE}}$ ) calculated from the average results of 40 independent trials, each trial having 200 generations for the three different weight sets. It is evident from the plot that the weightage set of  $w_1 = 0.9$  and  $w_2 = 0.1$  shows better performance with higher  $\eta_{\text{LAE}}$  values. The best combination is marked by bold font in Table 3.

The statistical measures like mean, best, worst, and standard deviation are calculated from 40 independent trials. Statistical performances are given in Table 4.

The results of the proposed PAHS approach are compared with the previous solutions reported in literature. Yıldız [24] used Hybrid Immune-Hill Climbing algorithm for solving this problem. Osyczka and Kundu [21] used distance method in GA to solve this problem and compared their results with that of the plain stochastic method. Sabarinath et al. [59] used fast and elitist NSGA II algorithm for solving this disc brake problem. Table 5 compares the results obtained by various

TABLE 3: Optimal design parameters of disc brake at different weights.

$W_1$	$W_2$	$X_1$ (mm)	$X_2$ (mm)	$X_3$ (N)	$X_4$	Mass of the brake (Kg)	Stopping time (Sec)	Comb. fitness	$\eta_{LAE}$
1	0	55	75	2666.9	2	0.1274	21.2723	$4.79E - 05$	0.215715
<b>0.9</b>	<b>0.1</b>	<b>79.9987</b>	<b>100.0002</b>	<b>2997</b>	<b>4</b>	<b>0.5117</b>	<b>6.1905</b>	<b>0.151448</b>	<b>0.399551</b>
0.8	0.2	79.9841	99.9841	3000	6	0.8539	4.1348	0.239908	0.29855
0.7	0.3	79.9880	99.9887	3000	8	1.1792	3.1422	0.29341	0.227969
0.6	0.4	79.9900	99.9901	3000	10	1.5099	2.5261	0.321191	0.180646
0.5	0.5	79.9885	99.9907	3000	11	1.7631	2.1963	0.310603	0.155044
0.4	0.6	79.9960	99.9964	2997	11	1.7641	2.1960	0.24995	0.154951
0.3	0.7	79.9888	99.9942	3000	11	1.7631	2.1968	0.189033	0.155041
0.2	0.8	79.9934	100.0140	2997	11	1.7686	2.1949	0.128576	0.154535
0.1	0.9	79.9970	109.0948	2999	11	2.6382	2.0885	0.095252	0.101438
0	1	80	110	3000	11	2.7890	2.071	1	0.095732

TABLE 4: Statistical performance for disc brake design optimization.

$W_1$	$W_2$	Best	Worst	Mean	Standard deviation
1	0	0.127401474	0.127433238	0.127409073	$8.12523E - 06$
0.9	0.1	1.079576294	1.079820227	1.079606351	$4.90341E - 05$
0.8	0.2	1.509991256	1.510078349	1.510009705	$1.79996E - 05$
0.7	0.3	1.768104744	1.768180817	1.768123371	$1.82142E - 05$
0.6	0.4	1.916348799	1.916466239	1.916370027	$2.06882E - 05$
0.5	0.5	1.979631	1.980991	1.979855	0.000272
0.4	0.6	2.022804	2.023961	2.023146	0.000294
0.3	0.7	2.065902	2.067994	2.066419	0.000422
0.2	0.8	2.109074	2.110377	2.109351	0.0003
0.1	0.9	2.14321361	2.143577898	2.143269642	$6.87429E - 05$
0	1	2.071118	2.072861	2.071514	0.000387

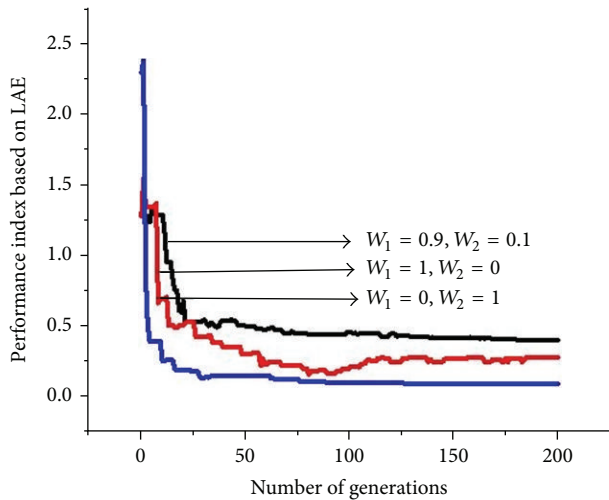


FIGURE 4: Comparison of results of disc brake at different weights.

methods including the proposed approach for solving the biobjective disc brake problem. It is evident from Table 5 that the results provided by the proposed PAHS approach are better than the results of Yıldız [24] and Osyczka and Kundu [21] with less function evaluations. Yıldız [24] also reported the extreme points (i.e., minimum value points for each separate

criterion) obtained from three methods in their paper. The extreme points reported for NSGA II algorithm is the same as that of the proposed PAHS algorithm. Comparison of the extreme points obtained by PAHS against other four methods is given in Table 5.

**5.2. Compression Coil Spring Problem.** The weightage set is varied in the range of 0 to 1 with a step size of 0.1. It is evident from the results that single objective optimization is achieved, when the weights are defined at their extreme values. The optimal design parameters of spring at different weights are shown in Table 6. When  $w_1 = 1$  and  $w_2 = 0$ , minimizing the volume of the spring is the effective objective function and it can be observed that the volume of the spring is minimum. In the other extreme, when  $w_1 = 0$  and  $w_2 = 1$ , maximizing strain energy stored is the effective objective function in which the strain energy stored is maximum. Here also the concept of performance index based on least average error (LAE) as explained in disc brake problem is applied. Based on the highest  $\eta_{LAE}$  value observed in Table 6, the weightage set of  $w_1 = 0.5$  and  $w_2 = 0.5$  is chosen as the best combination. Figure 5 shows the plot of performance index based on LAE ( $\eta_{LAE}$ ) calculated from the average results of 40 independent trials, each trial having 200 generations for the three different weight sets. It is evident from the plot that the weightage set of  $w_1 = 0.5$  and  $w_2 = 0.5$  shows better performance with higher



TABLE 5: Comparison of results for disc brake problem.

Method	Minima	$X = [x_1, x_2, x_3, x_4]^T$	$f(x) = [f_1(x), f_2(x)]^T$	$g(x) = [g_1(x), g_2(x), \dots, g_5(x)]^T$
Plain stochastic method	Min $f_1(x)$	[62.6, 83.5, 2920.9, 11]	[ <b>1.79</b> , 2.77]	[0.89, 0.00, 0.09, 0.76, 45.0]
	Min $f_2(x)$	[70.4, 106.6, 2948.4, 11]	[3.76, <b>2.24</b> ]	[16.1, 0.00, 0.25, 0.86, 264.0]
GA method	Min $f_1(x)$	[65.8, 86.1, 2982.4, 10]	[ <b>1.66</b> , 2.87]	[0.65, 0.25, 0.09, 0.75, 11.6]
	Min $f_2(x)$	[78.7, 108.3, 2988.3, 11]	[3.25, <b>2.11</b> ]	[9.57, 0.00, 0.22, 0.83, 340.0]
Hybrid Immune-Hill Climbing algorithm (HIHC)	Min $f_1(x)$	NA	[ <b>0.137</b> , 25.87]	NA
	Min $f_2(x)$	NA	[2.816, <b>2.083</b> ]	NA
NSGA II	Min $f_1(x)$	[55, 75, 2736.72, 2]	[ <b>0.1274</b> , 16.83]	[0, 22.5, 0.034, 0.75, 14122]
	Min $f_2(x)$	[79.99, 109.99, 2999.99, 11]	[3.3459, <b>2.071</b> ]	[10, 0, 0.23, 0.832, 361.2]
Proposed PAHS	Min $f_1(x)$	[57.95, 78.57, 2736.72, 2]	[ <b>0.1274</b> , 17.38]	[0.6142, 22.5, 0.09, 0.77, 1412.2]
	Min $f_2(x)$	[79.99, 109.99, 2999.99, 11]	[3.3459, <b>2.071</b> ]	[10, 0, 0.23, 0.832, 361.2]

TABLE 6: Optimal design parameters of compression coil spring at different weights.

$W_1$	$W_2$	$X_1$ (mm)	$X_2$ (mm)	$X_3$	Volume (mm <sup>3</sup> )	Strain energy stored (J)	Comb. fitness	$\eta_{LAE}$
1	0	7.1882	31.07	9	43567	100.926	0.000272	1.050567
0.9	0.1	7.1882	31.915	10	48776	119.617	0.107336	1.273232
0.8	0.2	7.1882	36.4877	9	51118	163.5858	0.128889	2.071811
0.7	0.3	7.7978	41.9044	7	56533	139.2175	0.239447	1.425222
0.6	0.4	7.7978	27.1831	9	69942	185.6148	0.239112	1.688518
<b>0.5</b>	<b>0.5</b>	<b>7.7978</b>	<b>53.92</b>	<b>8</b>	<b>80809</b>	<b>338.443</b>	<b>0.116524</b>	<b>2.777167</b>
0.4	0.6	8.4074	37.96	12	92581	131.135	0.525667	0.847559
0.3	0.7	8.4074	46.352	10	96911	199.013	0.308506	1.178813
0.2	0.8	9.1948	56.46	10	141201	251.755	0.239791	0.831192
0.1	0.9	9.1948	48.84	18	203591	292.851	0.159556	0.544499
0	1	7.7978	35.6234	24	138822	292.851	0.066173	0.914756

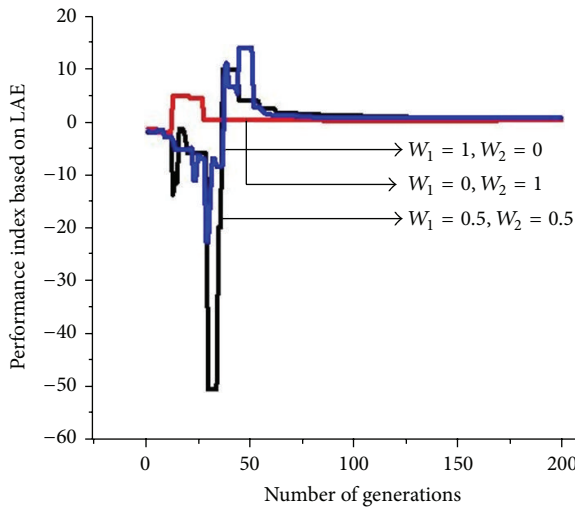


FIGURE 5: Comparison of results of biobjective helical spring problem.

$\eta_{LAE}$  values. The best combination is marked by bold font in Table 6.

The statistical measures like mean, best, worst, and standard deviation are calculated from 40 independent trials. Statistical performances are given in Table 7.

Since we are proposing a new biobjective design optimization for spring problem by converting the standard

single objective benchmark problem, we are unable to compare the results as in the case of disc brake. But when  $w_1 = 1$  and  $w_2 = 0$ , minimizing the volume of the spring is the effective objective function and thus it becomes single objective benchmark problem. The results obtained for single objective, that is, volume minimization for the proposed PAHS approach, are compared with that of the results reported in literature in Table 8. Since the values for comparison are available in inches, the result of the proposed PAHS algorithm is reported in inches as given in Table 8.

For validating our results for biobjective spring problem, the same problem is solved using NSGA II algorithm with the following values of the parameters of NSGA-II algorithm.

Variable type = real variable, population size = 50, crossover probability = 0.8, real-parameter mutation probability = 0.01, real-parameter SBX parameter = 10, real-parameter mutation parameter = 100, and total number of generations = 100. The results are presented in Table 9. It is evident from Table 9 that the results provided by the proposed PAHS approach are slightly better than the results of NSGA II algorithm.

### 6. Conclusion

In this paper, a recently proposed parameter adaptive harmony search algorithm based on the combinations of linear and exponential changes has been applied successfully to

TABLE 7: Statistical performance for compression coil spring optimization.

$W_1$	$W_2$	Best	Worst	Mean	Standard deviation
1	0	43567.2821	43568.3547	43567.8357	0.0006548
0.9	0.1	43898.3784	43899.5618	43898.9832	0.0004875
0.8	0.2	40894.3587	40896.3241	40895.5897	0.0002971
0.7	0.3	39573.0857	39574.9854	39574.0658	0.0002184
0.6	0.4	41965.1857	41966.5267	41965.7364	0.0003654
0.5	0.5	40404.4875	40405.2148	40404.9587	0.000238
0.4	0.6	37031.9547	37032.6874	37032.0548	0.000322
0.3	0.7	29073.3154	29074.6257	29073.9658	0.000483
0.2	0.8	28240.1586	28241.6328	28240.8294	0.000365
0.1	0.9	20359.0587	20360.8724	20360.0875	0.000587
0	1	3.4145E - 06	3.42654E - 06	3.4196E - 06	5.826489E - 06

TABLE 8: Comparison of optimal solutions for the compression coil spring design problem.

Variables/functions	Sandgren [15]	Chen and Tsao [16]	Wu and Chow [17]	Guo et al. [18]	Lampinen and Zelinka [19]	PSO of Kennedy and Eberhart [20]	Proposed PAHS
$D$ (in.)	1.180701	1.2287	1.227411	1.223	1.223042	1.223047	1.22304142
$N$	10	9	9	9	9	9	9
$d$ (in.)	0.283	0.283	0.283	0.283	0.283	0.283	0.283
$f_1(\mathbf{x})$ (in <sup>3</sup> )	2.7995	2.6709	2.6681	2.659	2.65856	2.658573	<b>2.658559</b>

TABLE 9: Optimum results of biobjective spring design optimization problem.

Method	Solution	$X_1$ (mm)	$X_2$ (mm)	$X_3$	Volume (mm <sup>3</sup> )	Strain energy stored (J)
NSGA II	Min. volume	7.1882	31.122112	9	<b>43645.734</b>	101.435
	Max. energy	7.7978	49.408799	10	88954.819	<b>325.6445</b>
Proposed weighted sum PAHS	Min. volume	7.1882	31.0654446	9	<b>43566.263</b>	100.883
	Max. energy	7.7978	53.91658	8	80892.1	<b>338.523</b>

solve multiobjective design optimization problems using weighted sum approach. In this weighted sum approach, a performance index based on least average error (LAE) was used to find the performance of each weightage set. Based on the higher LAE value, optimum weightage selection was determined. First, a standard benchmark problem of biobjective disc brake optimization has been solved and the results are compared with that of a plain stochastic method, simple genetic algorithm, HIHC, and NSGA II. Based on the analysis carried out, the weightage set of  $w_1 = 0.9$  and  $w_2 = 0.1$  is determined as the best selection for disc brake optimization problem. It is evident from the results that this approach performs better than other algorithms. Hence this approach is extended to solve a newly formulated biobjective design optimization of spring problem and the results are presented. The optimal weightage set of  $w_1 = 0.5$  and  $w_2 = 0.5$  is determined as the best selection for spring optimization problem. The result of spring problem is compared with the results of NSGA II algorithm and the results are better than NSGA II. This weighted sum approach based on LAE can be extended to any metaheuristic algorithm for implementing multiobjective optimization. Also Pareto optimal solutions can be found by using adaptive weighted sum method as a future extension of the present problem.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

- [1] J. L. Marcelin, "Genetic optimisation of gears," *International Journal of Advanced Manufacturing Technology*, vol. 17, no. 12, pp. 910–915, 2001.
- [2] K. Deb and S. Jain, "Multi-speed gearbox design using multi-objective evolutionary algorithms," *Journal of Mechanical Design*, vol. 125, no. 3, pp. 609–619, 2003.
- [3] H. Hirani, K. Athre, and S. Biswas, "Comprehensive design methodology for an engine journal bearing," *Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology*, vol. 214, no. 4, pp. 401–412, 2000.
- [4] J. S. Rao and R. Tiwari, "Optimum design and analysis of thrust magnetic bearings using multi objective genetic algorithms," *International Journal for Computational Methods in Engineering Science and Mechanics*, vol. 9, no. 4, pp. 223–245, 2008.
- [5] B. R. Rao and R. Tiwari, "Optimum design of rolling element bearings using genetic algorithms," *Mechanism and Machine Theory*, vol. 42, no. 2, pp. 233–250, 2007.
- [6] D.-H. Choi and K.-C. Yoon, "A design method of an automotive wheel-bearing unit with discrete design variables using genetic

- algorithms," *Transactions of ASME, Journal of Tribology*, vol. 123, no. 1, pp. 181–187, 2001.
- [7] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, December 1995.
- [8] P. Sabarinath, M. R. Thansekhar, and R. Saravanan, "Performance evaluation of particle swarm optimization algorithm for optimal design of belt pulley system," in *Swarm, Evolutionary, and Memetic Computing*, vol. 8297 of *Lecture Notes in Computer Science*, pp. 601–616, Springer, Cham, Switzerland, 2013.
- [9] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimization," *International Journal of Bio-Inspired Computation*, vol. 2, no. 2, pp. 78–84, 2010.
- [10] X. S. Yang and S. Deb, "Cuckoo Search via Lévy flights," in *Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC '09)*, pp. 210–214, IEEE, Coimbatore, India, December 2009.
- [11] 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.
- [12] V. Kumar, J. K. Chhabra, and D. Kumar, "Parameter adaptive harmony search algorithm for unimodal and multimodal optimization problems," *Journal of Computational Science*, vol. 5, no. 2, pp. 144–155, 2014.
- [13] K. Naidu, H. Mokhlis, and A. H. A. Bakar, "Multiobjective optimization using weighted sum Artificial Bee Colony algorithm for Load Frequency Control," *International Journal of Electrical Power & Energy Systems*, vol. 55, pp. 657–667, 2014.
- [14] A. Kattan, R. Abdullah, and R. A. Salam, "Harmony search based supervised training of artificial neural networks," in *Proceedings of the International Conference on Intelligent Systems, Modelling and Simulation (ISMS '10)*, pp. 105–110, Liverpool, UK, January 2010.
- [15] E. Sandgren, "Nonlinear integer and discrete programming in mechanical design optimization," *ASME Transactions on Mechanical Design*, vol. 112, no. 2, pp. 223–229, 1990.
- [16] J.-L. Chen and Y.-C. Tsao, "Optimal design of machine elements using genetic algorithms," *Journal of the Chinese Society of Mechanical Engineers*, vol. 14, no. 2, pp. 193–199, 1993.
- [17] S.-J. Wu and P.-T. Chow, "Genetic algorithms for nonlinear mixed discrete-integer optimization problems via meta-genetic parameter optimization," *Engineering Optimization*, vol. 24, no. 2, pp. 137–159, 1995.
- [18] C.-X. Guo, J.-S. Hu, B. Ye, and Y.-J. Cao, "Swarm intelligence for mixed-variable design optimization," *Journal of Zhejiang University Science*, vol. 5, no. 7, pp. 851–860, 2004.
- [19] J. Lampinen and I. Zelinka, "Mixed integer-discrete-continuous optimization by differential evolution," in *Proceedings of the 5th International Conference on Soft Computing*, pp. 71–76, Brno, Czech Republic, June 1999.
- [20] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp. 4104–4108, October 1997.
- [21] A. Osyczka and S. Kundu, "A modified distance method for multicriteria optimization, using genetic algorithms," *Computers and Industrial Engineering*, vol. 30, no. 4, pp. 871–882, 1996.
- [22] T. Ray and K. M. Liew, "A swarm metaphor for multiobjective design optimization," *Engineering Optimization*, vol. 34, no. 2, pp. 141–153, 2002.
- [23] A. R. Yıldız, N. Öztürk, N. Kaya, and F. Öztürk, "Hybrid multi-objective shape design optimization using Taguchi's method and genetic algorithm," *Structural and Multidisciplinary Optimization*, vol. 34, no. 4, pp. 317–332, 2007.
- [24] A. R. Yıldız, "An effective hybrid immune-hill climbing optimization approach for solving design and manufacturing optimization problems in industry," *Journal of Materials Processing Technology*, vol. 209, no. 6, pp. 2773–2780, 2009.
- [25] X.-S. Yang and S. Deb, "Multi objective cuckoo search for design optimization," *Computers and Operations Research*, vol. 40, no. 6, pp. 1616–1624, 2013.
- [26] X.-S. Yang, "Multiobjective firefly algorithm for continuous optimization," *Engineering with Computers*, vol. 29, no. 2, pp. 175–184, 2013.
- [27] X.-S. Yang, M. Karamanoglu, and X. Heb, "Multi-objective flower algorithm for optimization," *Procedia Computer Science*, vol. 18, pp. 861–868, 2013.
- [28] G. Reynoso-Meza, X. Blasco, J. Sanchis, and J. M. Herrero, "Comparison of design concepts in multi-criteria decision-making using level diagrams," *Information Sciences*, vol. 221, pp. 124–141, 2013.
- [29] K. Deb and M. Goyal, "Optimizing engineering designs using a combined genetic search," in *Proceedings of the 7th International Conference on Genetic Algorithms*, I. T. Back, Ed., pp. 512–528, 1997.
- [30] D. Datta and J. R. Figueira, "A real-integer-discrete-coded particle swarm optimization for design problems," *Applied Soft Computing Journal*, vol. 11, no. 4, pp. 3625–3633, 2011.
- [31] S. He, E. Prempan, and Q. H. Wu, "An improved particle swarm optimizer for mechanical design optimization problems," *Engineering Optimization*, vol. 36, no. 5, pp. 585–605, 2004.
- [32] K. Deb, A. Pratap, and S. Moitra, "Mechanical component design for multiple objectives using elitist non-dominated sorting GA," Technical Report No. 200002, Kanpur Genetic Algorithms Laboratory (KanGAL), Indian Institute of Technology, Kanpur, India, 2000.
- [33] Z. W. Geem, J.-H. Kim, and G. V. Loganathan, "Harmony search optimization: application to pipe network design," *International Journal of Modelling and Simulation*, vol. 22, no. 2, pp. 125–133, 2002.
- [34] K. R. Paik, J. H. Jeong, and J. H. Kim, "Use of a harmony search for optimal design of coffer dam drainage pipes," *Journal of the Korean Society of Civil Engineers*, vol. 21, no. 2, pp. 119–128, 2001.
- [35] Z. W. Geem, "Optimal cost design of water distribution networks using harmony search," *Engineering Optimization*, vol. 38, no. 3, pp. 259–277, 2006.
- [36] Z. Geem and H. Hwangbo, "Application of harmony search to multi-objective optimization for satellite heat pipe design," in *Proceedings of the Us-Korea Conference on Science, Technology and Entrepreneurship*, pp. 1–3, Citeseer, Teaneck, NJ, USA, 2006.
- [37] S. O. Degertekin, "Optimum design of steel frames using harmony search algorithm," *Structural and Multidisciplinary Optimization*, vol. 36, no. 4, pp. 393–401, 2008.
- [38] L. D. S. Coelho and V. C. Mariani, "An improved harmony search algorithm for power economic load dispatch," *Energy Conversion and Management*, vol. 50, no. 10, pp. 2522–2526, 2009.
- [39] V. R. Pandi, B. K. Panigrahi, M. K. Mallick, A. Abraham, and S. Das, "Improved harmony search for economic power dispatch," in *Proceedings of the 9th International Conference on Hybrid Intelligent Systems (HIS '09)*, pp. 403–408, Shenyang, China, August 2009.

- [40] S. Sivasubramani and K. S. Swarup, "Multi-objective harmony search algorithm for optimal power flow problem," *International Journal of Electrical Power and Energy Systems*, vol. 33, no. 3, pp. 745–752, 2011.
- [41] Z. W. Geem, K. S. Lee, and Y. Park, "Application of harmony search to vehicle routing," *American Journal of Applied Sciences*, vol. 2, no. 12, pp. 1552–1557, 2005.
- [42] Z. W. Geem, C. Tseng, and Y. Park, "Harmony search for generalized orienteering problem: best touring in China," in *Advances in Natural Computation*, vol. 3612 of *Lecture Notes in Computer Science*, pp. 741–750, Springer, Berlin, Germany, 2005.
- [43] H. Xu, X. Z. Gao, T. Wang, and K. Xue, "Harmony search optimization algorithm: application to a reconfigurable mobile robot prototype," in *Recent Advances in Harmony Search Algorithm*, vol. 270 of *Studies in Computational Intelligence*, pp. 11–22, Springer, Berlin, Germany, 2010.
- [44] B. Amiri, L. Hossain, and S. E. Mosavi, "Applications of harmony search algorithm on clustering," in *Proceedings of the World Congress on Engineering and Computer Science*, pp. 460–465, 2010.
- [45] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1567–1579, 2007.
- [46] Z. Kong, L. Gao, L. Wang, Y. Ge, and S. Li, "On an adaptive harmony search algorithm," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 9, pp. 2551–2560, 2009.
- [47] M. G. H. Omran and M. Mahdavi, "Global-best harmony search," *Applied Mathematics and Computation*, vol. 198, no. 2, pp. 643–656, 2008.
- [48] S. Das, A. Mukhopadhyay, A. Roy, A. Abraham, and B. K. Panigrahi, "Exploratory power of the harmony search algorithm: analysis and improvements for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, vol. 41, no. 1, pp. 89–106, 2011.
- [49] O. M. Alia and R. Mandava, "The variants of the harmony search algorithm: an overview," *Artificial Intelligence Review*, vol. 36, no. 1, pp. 49–68, 2011.
- [50] M. Mahdavi and H. Abolhassani, "Harmony  $K$ -means algorithm for document clustering," *Data Mining and Knowledge Discovery*, vol. 18, no. 3, pp. 370–391, 2009.
- [51] C. Worasuchee, "A harmony search with adaptive pitch adjustment for continuous optimization," *International Journal of Hybrid Information Technology*, vol. 4, no. 4, pp. 13–24, 2011.
- [52] X. S. Yang, "Harmony search as a metaheuristic algorithm," in *Music Inspired Harmony Search Algorithm, Theory and Applications*, Z. W. Geem, Ed., pp. 1–14, Springer, Berlin, Germany, 2009.
- [53] N. Taherinejad, "Highly reliable harmony search algorithm," in *Proceedings of the European Conference on Circuit Theory and Design*, pp. 818–822, August 2009.
- [54] P. Chakraborty, G. G. Roy, S. Das, D. Jain, and A. Abraham, "An improved harmony search algorithm with differential mutation operator," *Fundamenta Informaticae*, vol. 95, no. 4, pp. 401–426, 2009.
- [55] C.-M. Wang and Y.-F. Huang, "Self-adaptive harmony search algorithm for optimization," *Expert Systems with Applications*, vol. 37, no. 4, pp. 2826–2837, 2010.
- [56] R. T. Marler and J. S. Arora, "The weighted sum method for multi-objective optimization: new insights," *Structural and Multidisciplinary Optimization*, vol. 41, no. 6, pp. 853–862, 2010.
- [57] S. Hemamalini and S. P. Simon, "Economic/emission load dispatch using artificial bee colony algorithm," *ACEEE International Journal on Electrical and Power Engineering*, vol. 1, no. 2, pp. 27–33, 2010.
- [58] Y. K. Jain and S. K. Bhandare, "Min max normalization based data perturbation method for privacy protection," *International Journal of Computer & Communication Technology*, vol. 2, 2011.
- [59] P. Sabarinath, R. Hariharasudhan, M. R. Thansekhar, and R. Saravanan, "Optimal design of disc brake using NSGA II algorithm," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 3, no. 3, pp. 1400–1405, 2014.