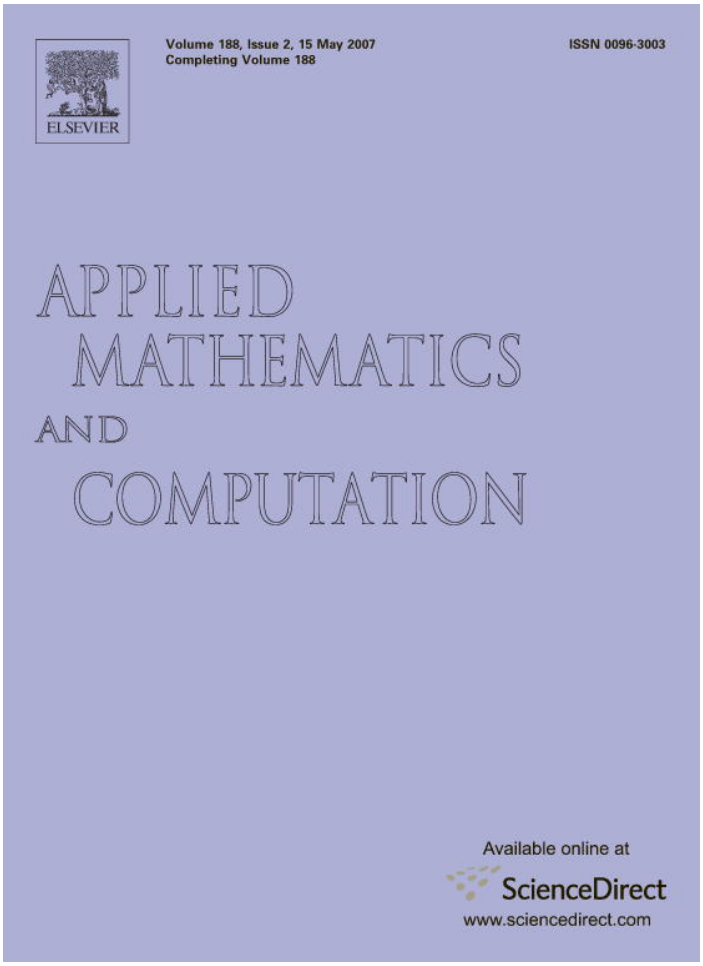

Available online at www.sciencedirect.com
ScienceDirect
Applied Mathematics and Computation 188 (2007) 1567–1579
APPLIED MATHEMATICS AND COMPUTATION
www.elsevier.com/locate/amc

An improved harmony search algorithm for solving optimization problems

M. Mahdavi [a], M. Fesanghary [b,*], E. Damangir [b]

[a] Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
[b] Department of Mechanical Engineering, Amirkabir University of Technology, Tehran 15875-4413, Iran

Abstract

This paper develops an Improved harmony search (IHS) algorithm for solving optimization problems. IHS employs a novel method for generating new solution vectors that enhances accuracy and convergence rate of harmony search (HS) algorithm. In this paper the impacts of constant parameters on harmony search algorithm are discussed and a strategy for tuning these parameters is presented. The IHS algorithm has been successfully applied to various benchmarking and standard engineering optimization problems. Numerical results reveal that the proposed algorithm can find better solutions when compared to HS and other heuristic or deterministic methods and is a powerful search algorithm for various engineering optimization problems.

Keywords: Global optimization; Heuristics; Harmony search algorithm; Mathematical programming

1. Introduction

A meta-heuristic algorithm, mimicking the improvisation process of music players, has been recently developed and named harmony search (HS) [1]. harmony search algorithm had been very successful in a wide variety of optimization problems [2–6], presenting several advantages with respect to traditional optimization techniques such as the following [6]: (a) HS algorithm imposes fewer mathematical requirements and does not require initial value settings of the decision variables. (b) As the HS algorithm uses stochastic random searches, derivative information is also unnecessary. (c) The HS algorithm generates a new vector, after considering all of the existing vectors, whereas the genetic algorithm (GA) only considers the two parent vectors. These features increase the flexibility of the HS algorithm and produce better solutions.

HS is good at identifying the high performance regions of the solution space at a reasonable time, but gets into trouble in performing local search for numerical applications. In order to improve the fine-tuning characteristic of HS algorithm, IHS employs a new method that enhances fine-tuning characteristic and

* Corresponding author.
E-mail addresses: mehrdad.mahdavi@gmail.com (M. Mahdavi), fesanghary@gmail.com (M. Fesanghary), damangir@aut.ac.ir (E. Damangir).

0096-3003/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2006.11.033

convergence rate of harmony search. The IHS algorithm has the power of the HS algorithm with the fine-tuning feature of mathematical techniques and can outperform either one individually.

To show the great power of this method, IHS algorithm is applied to various standard engineering optimization problems. Numerical results reveal that the proposed algorithm is a powerful search algorithm for various optimization problems.

2. Improved harmony search algorithm

This section describes the proposed improved harmony search (IHS) algorithm. First, a brief overview of the HS is provided, and finally the modification procedures of the proposed IHS algorithm are stated.

2.1. Harmony search algorithm

Harmony search (HS) algorithm was recently developed in an analogy with music improvisation process where music players improvise the pitches of their instruments to obtain better harmony [6]. The steps in the procedure of harmony search are shown in Fig. 1. They are as follows [6]:

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.

2.1.1. Initialize the problem and algorithm parameters

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

$$\text{Minimize} \quad f(x) \text{ subject to } \quad x_i \in Xi = 1, 2, \ldots, N, \tag{1}$$

where $f(x)$ is an objective function; $\mathbf{x}$ is the set of each decision variable x_i; N is the number of decision variables, X_i is the set of the possible range of values for each decision variable, that is ${}_{\mathrm{L}}x_i \leqslant X_i \leqslant {}_{\mathrm{U}}x_i$ and ${}_{\mathrm{L}}x_i$ and ${}_{\mathrm{U}}x_i$ are the lower and upper bounds for each decision variable. The HS algorithm 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. This HM is similar to the genetic pool in the GA [3]. Here, HMCR and PAR are parameters that are used to improve the solution vector. Both are defined in Step 3.

2.1.2. Initialize the harmony memory

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

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

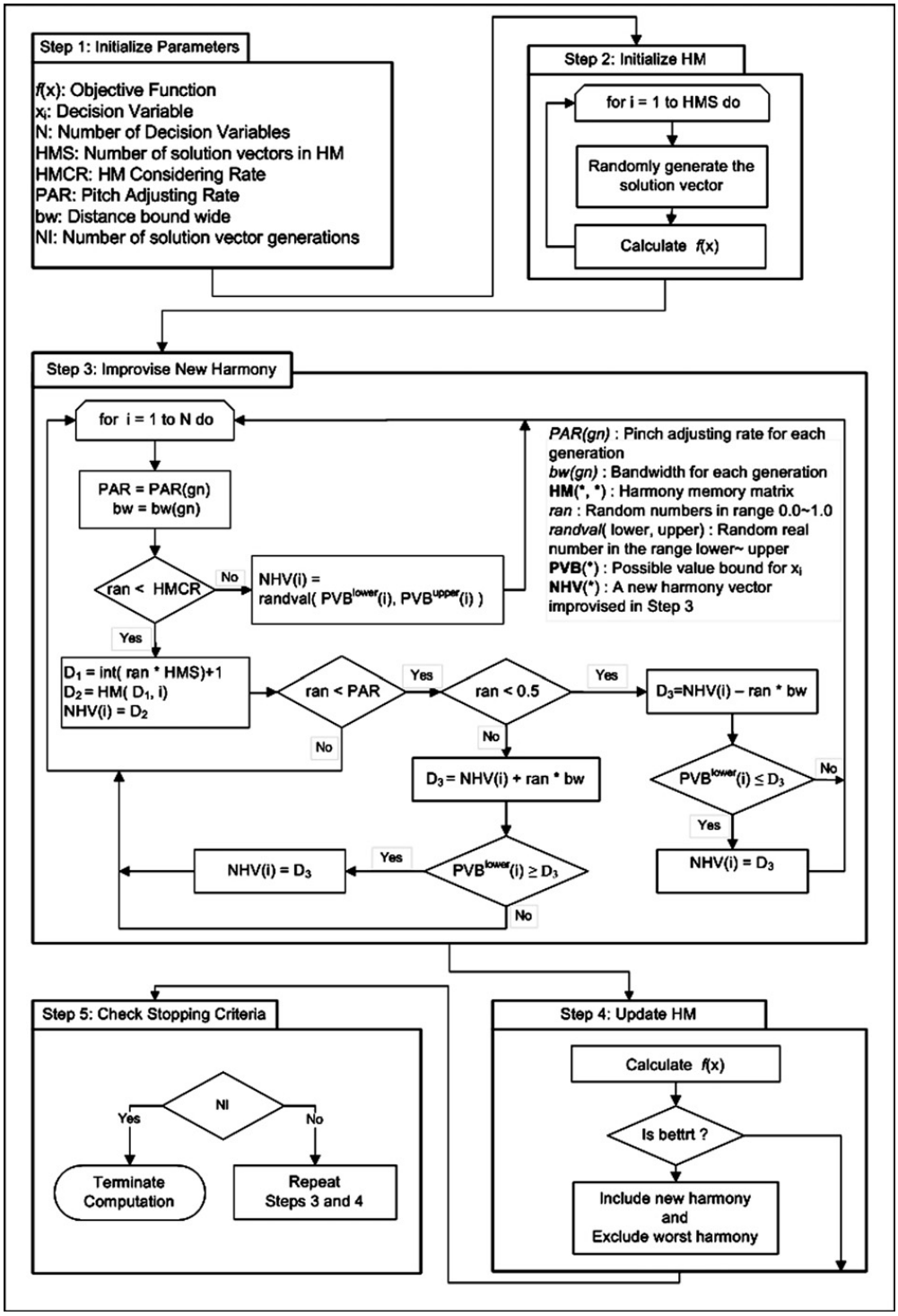

Fig. 1. Optimization procedure of the improved harmony search algorithm.

2.1.3. Improvise a new harmony

A new harmony vector, $x' = (x'_1, x'_2, \ldots, x'_N)$, is generated based on three rules: (1) memory consideration, (2) pitch adjustment and (3) random selection. Generating a new harmony is called 'improvisation' [6].

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 $(x'^1_1 - x'^{HMS}_1)$. 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.

$$x_i' \leftarrow \begin{cases} x_i' \in \{x_i^1, x_i^2, \ldots, x_i^{\mathrm{HMS}}\} & \text{with probability} \quad \mathrm{HMCR}, \\ x_i' \in X_i & \text{with probability} \quad (1 - \mathrm{HMCR}). \end{cases} \tag{3}$$

For example, a HMCR of 0.85 indicates that the HS algorithm will choose the decision variable value from historically stored values in the HM with an 85% probability or from the entire possible range with a (100–85)% 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:

$$\text{Pitch adjusting decision for } x_i' \leftarrow \begin{cases} \text{Yes} & \text{with probability} \quad \mathrm{PAR}, \\ \text{No} & \text{with probability} \quad (1 - \mathrm{PAR}). \end{cases} \tag{4}$$

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 follow:

$$x_i' \leftarrow x_i' \pm \mathrm{rand}() * \mathrm{bw}, \tag{5}$$

where
bw is an arbitrary distance bandwidth
rand( ) is a random number 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.

2.1.4. Update harmony memory

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

2.1.5. Check stopping criterion

If the stopping criterion (maximum number of improvisations) is satisfied, computation is terminated. Otherwise, Steps 3 and 4 are repeated.

2.2. Proposed method

The HMCR and PAR parameters introduced in Step 3 help the algorithm find globally and locally improved solutions, respectively [6].

PAR and bw in HS algorithm are very important parameters in fine-tuning of optimized solution vectors, and can be potentially useful in adjusting convergence rate of algorithm to optimal solution. So fine adjustment of these parameters are of great interest.

The traditional HS algorithm uses fixed value for both PAR and bw. In the HS method PAR and bw values adjusted in initialization step (Step 1) and cannot be changed during new generations. The main drawback of this method appears in the number of iterations the algorithm needs to find an optimal solution.

Small PAR values with large bw values can cause to poor performance of the algorithm and considerable increase in iterations needed to find optimum solution. Although small bw values in final generations increase the fine-tuning of solution vectors, but in early generations bw must take a bigger value to enforce the algorithm to increase the diversity of solution vectors. Furthermore large PAR values with small bw values usually cause the improvement of best solutions in final generations which algorithm converged to optimal solution vector.

The key difference between IHS and traditional HS method is in the way of adjusting PAR and bw. To improve the performance of the HS algorithm and eliminate the drawbacks lies with fixed values of PAR

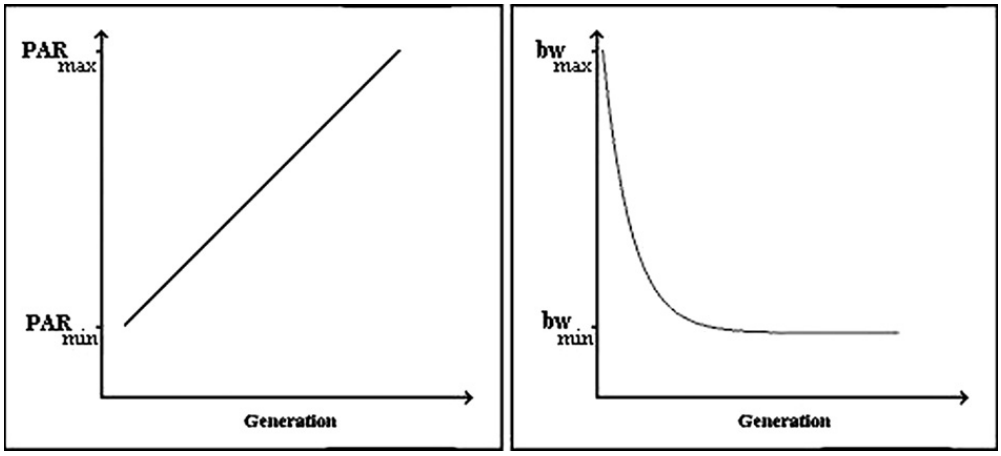

Fig. 2. Variation of PAR and bw versus generation number.

and bw, IHS algorithm uses variables PAR and bw in improvisation step (Step 3). PAR and bw change dynamically with generation number as shown in Fig. 2 and expressed as follow:

$$\mathrm{PAR}(\mathrm{gn}) = \mathrm{PAR}_{\min} + \frac{(\mathrm{PAR}_{\max} - \mathrm{PAR}_{\min})}{NI} \times \mathrm{gn}, \tag{6}$$

where
PAR pitch adjusting rate for each generation
$\mathrm{PAR}_{\min}$ minimum pitch adjusting rate
$\mathrm{PAR}_{\max}$ maximum pitch adjusting rate
NI number of solution vector generations
gn generation number

and

$$\mathrm{bw}(\mathrm{gn}) = \mathrm{bw}_{\max} \exp(c \cdot \mathrm{gn}),$$
$$c = \frac{\mathrm{Ln}\left(\frac{\mathrm{bw}_{\min}}{\mathrm{bw}_{\max}}\right)}{\mathrm{NI}}, \tag{7}$$

where
bw(gn) bandwidth for each generation
$\mathrm{bw}_{\min}$ minimum bandwidth
$\mathrm{bw}_{\max}$ maximum bandwidth

3. Examples

Several examples taken from the optimization literature used to show the way in which the proposed approach works. These examples have been previously solved using a variety of other techniques, which is useful to show the validity and effectiveness of the proposed algorithm. The IHS algorithm parameters for all examples presented in this paper are shown in Table 1.

3.1. Constrained function I: minimization of the weight of spring

This problem is described by Arora [7], Coello [8] and Belegundu [9]. It consists of minimizing the weight $(f(\vec{x}))$ of a tension/compression spring subject to constraints on shear stress, surge frequency and minimum

Table 1
Improved harmony search parameters used for test problems

Problem	HMCR	PAR		HMS	bw		NI
		PAR_{min}	PAR_{max}		bw_{min}	bw_{max}	
Minimization of the weight of spring	0.95	0.35	0.99	4	5e−4	0.05	50,000
Pressure vessel design(4 inequalities)	0.95	0.45	0.99	6	1e−4	20	200,000
Pressure vessel design(6 inequalities)	0.95	0.45	0.99	6	1e−5	4	200,000
Welded beam design	0.95	0.45	0.99	8	5e−4	2.5	300,000
Disjoint feasible region	0.95	0.35	0.99	4	1e−5	0.1	20,000
Constrained function V	0.95	0.45	0.99	5	1e−6	0.5	10,000
Unconstrained function I	0.95	0.35	0.99	7	1e−6	4	3000
Unconstrained function II	0.95	0.35	0.99	7	1e−6	4	6000

deflection as shown in Fig. 3. The design variables are the mean coil diameter $D(=x_1)$; the wire diameter $d(=x_2)$ and the number of active coils $N(=x_3)$. The problem can be stated as:

$$\text{Minimize} \quad f(\vec{x}) = (x_3+2)x_2x_1^2 \tag{8}$$

$$\begin{aligned}\text{Subject to} \quad & g_1(\vec{x}) = 1 - \frac{x_2^3 x_3}{71785x_1^4} \leqslant 0,\\ & g_2(\vec{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leqslant 0,\\ & g_3(\vec{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leqslant 0,\\ & g_4(\vec{x}) = \frac{x_2+x_1}{1.5} - 1 \leqslant 0.\end{aligned} \tag{9}$$

This problem has been solved by Belegundu [9] using eight different mathematical optimization techniques. Arora [7] also solved this problem using a numerical optimization technique called constraint correction at constant cost. Additionally, Coello [8] solved this problem using GA-based method. After 30,000 function evaluations the best solution is obtained at $x^* = (0.05115438, 0.34987116, 12.0764321)$ with corresponding function value equal to $f^*(x) = 0.0126706$. No constraints are active for this solution. Table 2 presents the best solution of this problem obtained using the IHS algorithm and compares the IHS results with solutions reported by other researchers. It is obvious from the Table 2 that the result obtained using IHS algorithm is better than those reported previously in the literature.

3.2. Constrained function II: pressure vessel design

A cylindrical vessel is capped at both ends by hemispherical heads as shown in Fig. 4. The objective is to minimize the total cost, including the cost of material, forming and welding. There are four design variables:

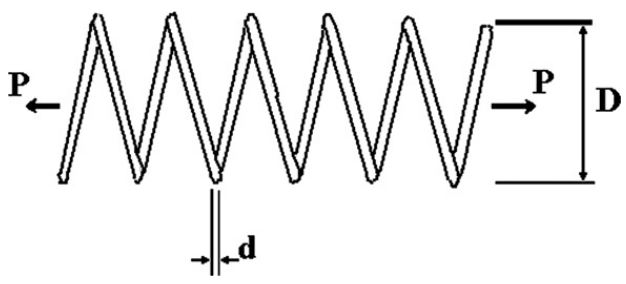

Fig. 3. Tension/compression spring.

Table 2
Optimal results for minimization of the weight of spring

Design variables	Best solution found			
	Proposed method	Arora [7]	Belegundu [9]	Coello [8]
$x_1(d)$	0.05115438	0.053396	0.050000	0.051989
$x_2(D)$	0.34987116	0.399180	0.315900	0.363965
$x_3(N)$	12.0764321	9.185400	14.25000	10.890522
$g_1(\vec{x})$	0.000000	0.000019	−0.000014	−0.000013
$g_2(\vec{x})$	−0.000007	−0.000018	−0.003782	−0.000021
$g_3(\vec{x})$	−4.0278401	−4.123832	−3.938302	−4.061338
$g_4(\vec{x})$	−0.7365723	−0.698283	−0.756067	−0.722698
$f(\vec{x})$	0.0126706	0.012730	0.012833	0.012681

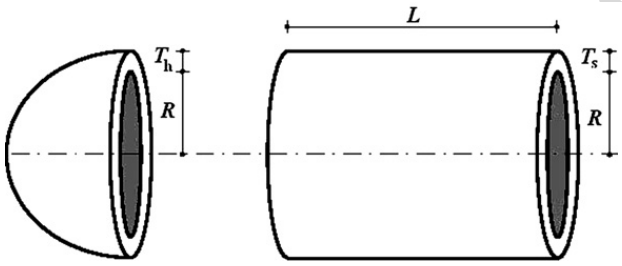

Fig. 4. Schematic of pressure vessel.

T_s (thickness of the shell, x_1), T_h (thickness of the head, x_2), R(inner radius, x_3) and L(length of cylindrical section of the vessel, not including the head, x_4). T_s and T_h are integer multiples of 0.0625 inch, witch are the available thickness of rolled steel plates, and R and L are continuous. Using the same notation given by Coello [10], the problem can be stated as follows:

$$\text{Minimize} \quad f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \tag{10}$$

$$\begin{aligned} \text{Subject to} \quad & g_1(\vec{x}) = -x_1 + 0.0193x_3 \leqslant 0, \\ & g_2(\vec{x}) = -x_2 + 0.00954x_3 \leqslant 0, \\ & g_3(\vec{x}) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1,296,000 \leqslant 0, \\ & g_4(\vec{x}) = x_4 - 240 \leqslant 0. \end{aligned} \tag{11}$$

This problem has been solved before by Deb and Gene [11] using Genetic Adaptive Search, by Kannan and Kramer [12] using an augmented Lagrangian Multiplier approach, and by Coello [13] using GA. The optimal solution is obtained at $x^* = (0.75, 0.375, 38.86010, 221.36553)$ with corresponding function value equal to $f^*(x) = 5849.76169$. The comparisons of results are shown in Table 3. The results obtained using the IHS algorithm, were better optimized than any other earlier solutions reported in the literature.

Another variation of this problem, that has two extra inequalities (Eq. (12)), has been

$$\begin{aligned} & g_5(\vec{x}) = 1.1 - x_1 \leqslant 0, \\ & g_6(\vec{x}) = 0.6 - x_2 \leqslant 0 \end{aligned} \tag{12}$$

Table 3
Optimal results for pressure vessel design (four inequalities)

Results	Proposed method	Deb and Gene [11]	Kannan and Kramer [12]	Coello [13]
Best	5849.7617	6410.3811	7198.0428	6069.3267

Table 4
Optimal results for pressure vessel design (six inequalities)

Optimal design variables	Proposed method	Wu and Chow [14]	Lee and Geem [6]	Sandgren [15]
$T_s(=x_1)$	1.125	1.125	1.125	1.125
$T_h(=x_2)$	0.625	0.625	0.625	0.625
$R(=x_3)$	58.29015	58.1978	58.2789	48.97
$L(=x_4)$	43.69268	44.2930	43.7549	106.72
$g_1(x)$	0.00000	−0.00178	−0.00022	−0.1799
$g_2(x)$	−0.06891	−0.06979	−0.06902	−0.1578
$g_3(x)$	−2.01500	−974.3	−3.71629	+97.760
$g_4(x)$	−196.307	−195.707	−196.245	−133.28
Cost ($)	7197.730	7207.494	7198.433	7980.894

solved by Wu and Chow [14], using GA-based approach, by Lee and Geem [6], using HS algorithm and by Sandgren [15], using branch and bound method. Table 4 shows the best solution vector from IHS algorithm and also provides the results obtained by Wu and Chow [14], Sandgren [15], and Lee and Geem [6]. The IHS algorithm showed better results than other methods.

3.3. Constrained function III: welded beam design

The welded beam structure, shown in Fig. 5, is a practical design problem that has been often used as a benchmark for testing different optimization methods [16–20]. The objective is to find the minimum fabricating cost of the welded beam subject to constraints on shear stress (τ), bending stress (σ), buckling load (P_c), end deflection (δ), and side constraint. There are four design variables: $h(=x_1)$, $l(=x_2)$, $t(=x_3)$ and $b(=x_4)$. The mathematical formulation of the objective function $f(x)$, which is the total fabricating cost mainly comprised of the set-up, welding labor, and material costs, is as follows:

$$\text{Minimize} \quad f(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \tag{13}$$

$$\begin{aligned}
\text{Subject to} \quad & g_1(\vec{x}) = \tau(\vec{x}) - \tau_{\max} \leqslant 0, \\
& g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{\max} \leqslant 0, \\
& g_3(\vec{x}) = x_1 - x_4 \leqslant 0, \\
& g_4(\vec{x}) = \delta(\vec{x}) - \delta_{\max} \leqslant 0, \\
& g_5(\vec{x}) = P - P_C(\vec{x}) \leqslant 0, \\
& 0.125 \leqslant x_1 \leqslant 0, \quad 0.1 \leqslant x_2, \ \ x_3 \leqslant 10, \ \ 0.1 \leqslant x_4 \leqslant 5,
\end{aligned} \tag{14}$$

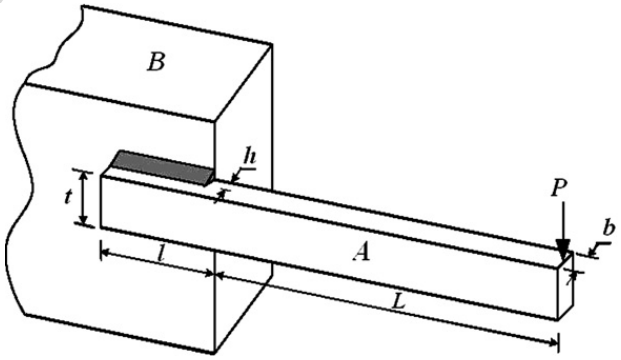

Fig. 5. Welded beam structure.

where

$$
\begin{aligned}
&\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2},\\
&\tau' = \frac{P}{\sqrt{2}x_1x_2}, \quad \tau'' = \frac{MR}{J}, \quad M = P\left(L + \frac{x_2}{2}\right),\\
&R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2},\\
&J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\},\\
&\sigma(\vec{x}) = \frac{6PL}{x_4x_3^2}, \quad \delta(\vec{x}) = \frac{6PL^3}{Ex_3^2x_4},\\
&P_C(\vec{x}) = \frac{4.013E\sqrt{x_3^2x_4^6/36}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),\\
&P = 6000\text{ lb}, \quad L = 14\text{ in.}, \quad \delta_{\max} = 0.25\text{ in.}, \quad E = 30 \times 10^6\text{ psi}, \quad G = 12 \times 10^6\text{ psi},\\
&\tau_{\max} = 13,600\text{ psi}, \quad \sigma_{\max} = 30,000\text{ psi}.
\end{aligned}
\tag{15}
$$

Deb [16,17] and Coello [13,21] solved this problem using GA-based methods. Radgsdell and Phillips [20] compared optimal results of different optimization methods that were mainly based on mathematical optimization algorithms. These methods, are APPROX (Griffith and Stewart's successive linear approximation), DAVID (Davidon–Fletcher–Powell with a penalty function), SIMPLEX (Simplex method with a penalty function), and RANDOM (Richardson's random method) algorithms. Lee and Geem [6] solved the problem using HS method. The comparison of results, are shown in Table 5.

The IHS result, which was obtained after approximately 200,000 searches, was better than those reported by Coello [21], who got the best results between others.

3.4. Constrained function IV: disjoint feasible region

This problem was originally proposed by Michalewicz and Schoenauer [22]. In this problem there are three design variables (x_1, x_2, x_3), one nonlinear inequality constraints and 6 boundary conditions. A point (x_1, x_2, x_3) is feasible if and only if there exist p, q and r such that below inequality holds. The problem can be stated as follows:

$$\text{Maximize} \quad f(\vec{x}) = \frac{100 - (x_1 - 5)^2 - (x_2 - 5)^2 - (x_3 - 5)^2}{100} \tag{16}$$

$$\text{Subject to} \quad g(\vec{x}) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625 \leqslant 0, \tag{17}$$

Table 5
Optimal results for welded beam design (N/A not available)

Methods	Optimal design variables (x)				Cost
	h	l	t	b	
Coello [13]	N/A	N/A	N/A	N/A	1.8245
Coello [21]	0.2088	3.4205	8.9975	0.2100	1.7483
Lee and Geem [6]	0.2442	6.2231	8.2915	0.2443	2.3807
Ragsdell and Phillips [20]					
APPROX	0.2444	6.2189	8.2915	0.2444	2.3815
DAVID	0.2434	6.2552	8.2915	0.2444	2.3841
SIMPLEX	0.2792	5.6256	7.7512	0.2796	2.5307
RANDOM	0.4575	4.7313	5.0853	0.6600	4.1185
Deb [16]	N/A	N/A	N/A	N/A	2.38
Deb [17]	0.2489	6.1730	8.1789	0.2533	2.4328
Proposed method	0.20573	3.47049	9.03662	0.20573	1.7248

where

$$0 \leqslant x_i \leqslant 10, \quad i = 1, 2, 3,$$
$$p, q, r = 1, 2, \ldots, 9.$$

This problem has been solved by Coello [8] using a GA-based method and by Koziel and Michalewicz using a GA with a technique called homomorphous maps [23].

The optimal solution is located at point $x^* = (5, 5, 5)$ with corresponding function value equal to $f^*(x) = 1$. Table 6 compares the best solution of example 3.4 obtained using the IHS algorithm with previous best solutions reported by Koziel and Michalewicz [23], and Coello [8]. Koziel and Michalewicz, and Coello obtained a best solution function value of $f(x) = 0.999999$ and $f(x) = 1.00000$, respectively, using the GA-based methods. Note that the approach proposed by Koziel and Michalewicz [23] required 1,400,000 evaluations of the fitness function to produce the result shown in Table 6. The approach of Coello [8] required 80,000 evaluations of the fitness function. In contrast, IHS only required 20,000 evaluations of the fitness function.

3.5. Constrained function V

The problem can be stated as follows:

$$\text{Minimize} \quad f(\vec{x}) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \tag{18}$$

$$\text{Subject to} \quad g_1(\vec{x}) = 4.84 - (x_1 - 0.05)^2 - (x_2 - 2.5)^2 \geqslant 0,$$
$$g_2(\vec{x}) = x_1^2 + (x_2 - 2.5)^2 - 4.84 \geqslant 0, \tag{19}$$
$$0 \leqslant x_1 \leqslant 6, \quad 0 \leqslant x_2 \leqslant 6.$$

The function have two design variables (x_1, x_2), two nonlinear inequality constraints and four boundary conditions. The unconstrained objective function has a minimum solution at (3, 2) with a function value of $f(\vec{x}) = 0$. However, due to the presence of constraints, this solution is no more feasible. The feasible region is a narrow crescent-shaped region (approximately 0.7% of the total search space) with the optimum solution lying on the second constraint, as shown in Fig. 6. The best solution is obtained at $x^* = (2.2468258, 2.381863)$ with corresponding function value equal to $f^*(x) = 13.590841$. No constraints are active for this solution. Table 7 presents the best solution of example 3.5 obtained using the IHS algorithm and compares the IHS results with solutions reported by Deb [16], using GA-based method and Lee and Geem [6], using HS method.

3.6. Unconstrained function I

$$\text{Minimize} \quad f(\vec{x}) = \exp\left\{\frac{1}{2}(x_1^2 + x_2^2 - 25)^2\right\} + \sin^4(4x_1 - 3x_2) + \frac{1}{2}(2x_1 + x_2 - 10). \tag{20}$$

The objective function $f(\vec{x})$ has a minimum solution at $x = (3, 4)$ with a corresponding function value of $f(\vec{x}) = 1.0$, as shown in Fig. 7. The two design variables x_1, x_2 were initially bounded between -50 and 50. The algorithm found the best solution at $x^* = (3.00000, 3.99999)$ with the corresponding objective function value of $f^*(x) = 1.00000$. The IHS solution is very close to the global optimum value.

Table 6
Comparison of results for disjoint feasible region (N/A not available)

Design variables	Best solution		
	Proposed method	Coello [8]	Koziel and Michalewics [23]
x_1	5.000000	5.0000	N/A
x_2	4.999999	5.0000	N/A
x_3	5.000001	5.0000	N/A
$f(\vec{x})$	0.9999999	1.000000	0.9999998

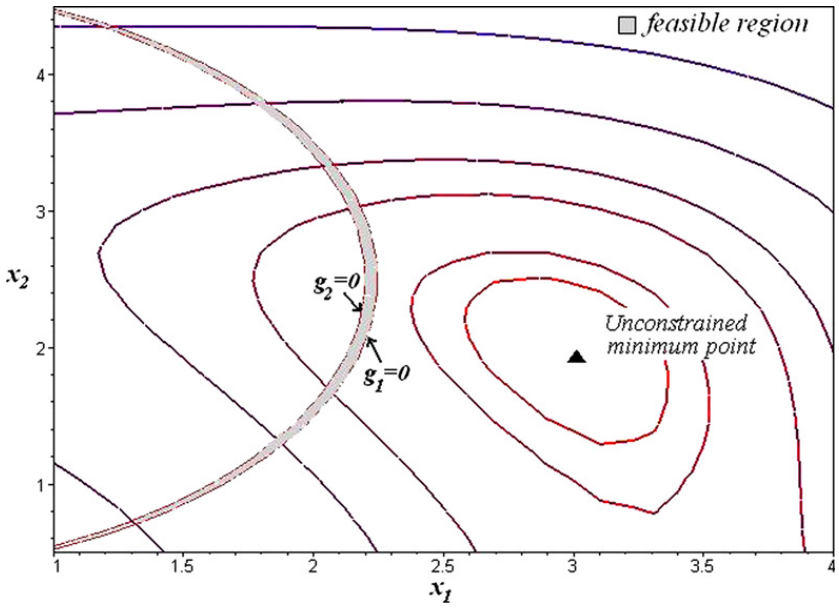

Fig. 6. Constrained function V.

Table 7
Optimal results for constrained function V (N/A not available)

Methods	Optimal design variables (x) and constraints (g)				$f(\vec{x})$
	x_1	x_2	g_1	g_2	
Deb [16]	N/A	N/A	N/A	N/A	13.59085
Lee and Geem [6]	2.246840	2.382136	0.00002	0.22218	13.590845
Proposed method	2.2468258	2.381863	0.00000	0.22218	13.590841

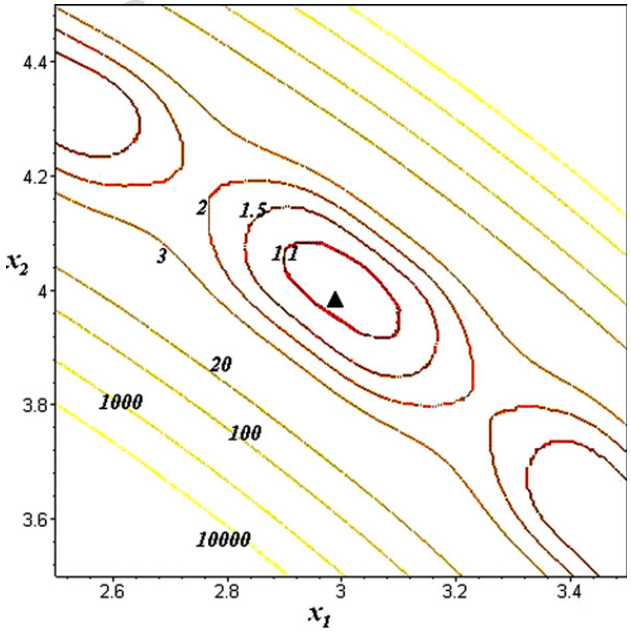

Fig. 7. Unconstrained function I.

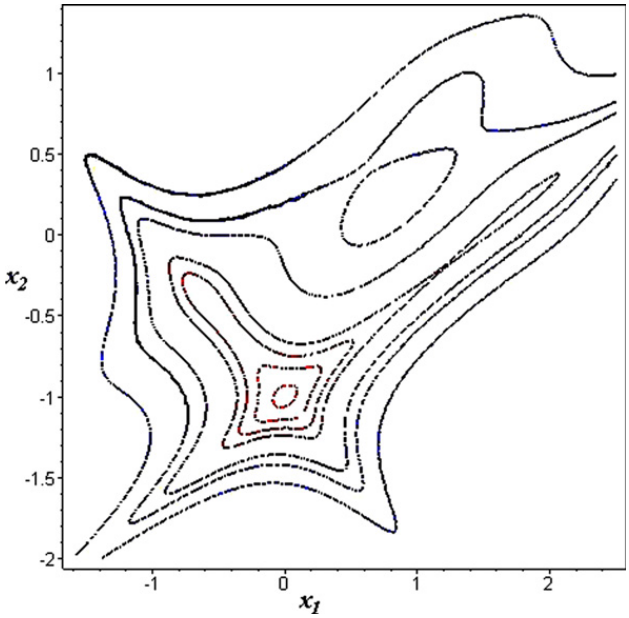

Fig. 8. Unconstrained function II.

3.7. Unconstrained function II

The problem can be stated as follows:

$$\begin{aligned}\text{Minimize}\quad f(\vec{x}) = {} & \{1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\} \\ & \times \{30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\}. \end{aligned} \tag{21}$$

This function is an eighth-order polynomial in two variables. As shown in Fig. 8, however, the function has four local minima, one of which is global, as follows [6]: $f(1.2, 0.8) = 840.0$, $f(1.8, 0.2) = 84.0$, $f(0.6, 0.4) = 30.0$ and $f^*(0, 1.0) = 3.0$ (global minimum). In this example, the bounds for two design variables (x_1 and x_2) were set between -50 and 50. After 2340 searches, the best solution vector found by the IHS algorithm was $x = (0.000000, -1.000001)$ with a corresponding function value of $f(x) = 3.000000$. The IHS solution is very close to the global optimum value.

4. Conclusion

This paper has introduced an improved harmony search algorithm which has the power of the HS algorithm with the fine-tuning feature of mathematical techniques. The impacts of constant parameters on harmony search algorithm were discussed and a strategy for improving the performances of HS algorithm through proper tuning these parameters was presented. IHS algorithm like harmony search algorithm is good at finding areas of the global optimum and is as good as mathematical techniques at fine-tuning within those areas. The proposed approach performed well in several test problems both in terms of the number of fitness function evaluations required and in terms of the quality of the solutions found. The results produced were compared against those generated with other (evolutionary and mathematical programming) techniques reported in the literature.

References

[1] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, Simulation 76 (2) (2001) 60–68.
[2] J.H. Kim, Z.W. Geem, E.S. Kim, Parameter estimation of the nonlinear Muskingum model using harmony search, J. Am. Water Resour. Assoc. 37 (5) (2001) 1131–1138.
[3] Z.W. Geem, J.H. Kim, G.V. Loganathan, Harmony search optimization: application to pipe network design, Int. J. Model. Simul. 22 (2) (2002) 125–133.

[4] S.L. Kang, Z.W. Geem, A new structural optimization method based on the harmony search algorithm, Comput. Struct. 82 (9–10) (2004) 781–798.
[5] Z.W. Geem, C. Tseng, Y. Park, Harmony search for generalized orienteering problem: best touring in China, in: Springer Lecture Notes in Computer Science, vol. 3412, 2005, pp. 741–750.
[6] K.S. Lee, Z.W. Geem, A new meta-heuristic algorithm for continues engineering optimization: harmony search theory and practice, Comput. Meth. Appl. Mech. Eng. 194 (2004) 3902–3933.
[7] J.S. Arora, Introduction to Optimum Design, McGraw-Hill, New York, 1989.
[8] C.A.C. Coello, Constraint-handling in genetic algorithms through the use of dominance-based tournament selection, Adv. Eng. Informatics 16 (2002) 193–203.
[9] A.D. Belegundu, A Study of Mathematical Programming Methods for Structural Optimization, PhD thesis, Department of Civil and Environmental Engineering, University of Iowa, Iowa, 1982.
[10] C.A.C. Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Comput. Meth. Appl. Mech. Eng. 191 (2002) 1245–1287.
[11] K. Deb, A.S. Gene, A robust optimal design technique for mechanical component design, in: D. Dasgupta, Z. Michalewicz (Eds.), Evolutionary Algorithms in Engineering Applications, Springer, Berlin, 1997, pp. 497–514.
[12] B.K. Kannan, S.N. Kramer, An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design, J. Mech. Des. Trans. ASME 116 (1994) 318–320.
[13] C.A.C. Coello, Constraint-handling using an evolutionary multiobjective optimization technique, Civ. Eng. Environ. Syst. 17 (2000) 319–346.
[14] S.J. Wu, P.T. Chow, Genetic algorithms for nonlinear mixed discrete-integer optimization problems via meta-genetic parameter optimization, Eng. Optim. 24 (1995) 137–159.
[15] E. Sandgren, Nonlinear integer and discrete programming in mechanical design optimization, J. Mech. Des. ASME 112 (1990) 223–229.
[16] K. Deb, An efficient constraint handling method for genetic algorithms, Comput. Meth. Appl. Mech. Eng. 186 (2000) 311–338.
[17] K. Deb, Optimal design of a welded beam via genetic algorithms, AIAA J. 29 (11) (1991).
[18] G.V. Reklaitis, A. Ravindran, K.M. Ragsdell, Engineering Optimization Methods and Applications, Wiley, New York, 1983.
[19] J.N. Siddall, Analytical Decision-Making in Engineering Design, Prentice-Hall, Englewood Cliffs, NJ, 1972.
[20] K.M. Ragsdell, D.T. Phillips, Optimal design of a class of welded structures using geometric programming, ASME J. Eng. Ind. Ser. B 98 (3) (1976) 1021–1025.
[21] C.A.C. Coello, Use of a self-adaptive penalty approach for engineering optimization problems, Comput. Ind. 41 (2) (2000) 113–127.
[22] Z. Michalewicz, M. Schoenauer, Evolutionary algorithms for constrained parameter optimization problems, Evol. Comput. 4 (1) (1996) 1–32.
[23] S. Koziel, Z. Michalewicz, Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization, Evol. Comput. 7 (1) (1999) 19–44.