A New Harmony Memory Updating Technique for Harmony Search Optimization Algorithm

Ali Rafiei Shahemabadi[a,*], Samsul Bahari B. Mohd. Noor[a], Mojtaba Alizadeh[b], Maryam Bt. Mohd Isa[a]

[a]Department of Electrical and Electronic Engineering, Faculty of Engineering, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor Darul Ehsan, Malaysia
[b]Faculty of Computer and Information Systems, Universiti Teknologi Malaysia, 81310 UTM Skudai, Malaysia

Abstract

The paper aims to introduce an innovative method for updating of the harmony memory. In this manner, the aim of the research is to create modification of the harmony search or HS algorithm. The potential advantage of this technique is that compared to the basic technique, the amount of evaluations pertaining to the HS algorithm for objective function of the HS algorithm are reduced significantly. The new method is superior to the old one in terms of maximum number of improvisations and subsequent to it, fast performance convergence. Compared to the older version, the new method was found to be more effective when assessed by four renowned functions usually used as test problems to detect performance among optimization algorithms.

Keywords: Harmony Search Algorithm, Evolutionary Algorithm, Optimization, Meta-Heuristic Algorithm

1. Introduction

When creating a balance between the best use of energy and achieving the desired results in performance, evolutionary algorithms are considered the best analyzing tools. The HS algorithm technique was introduced by [1] and since then remains one the preferred algorithm for optimization problems. The HS algorithm like other meta-heuristic algorithms employs high level techniques for exploration and exploitation of the huge solution space. Since the discovery of HS algorithm, it has been used extensively with positive results. Its applicability is universal, which is the reason for its high appeal. So far, the benefiting areas of human needs include pipe designing [2, 3], distribution networks for water [4], various structural designing [5, 6, 7], data clustering procedures [8], industrial designing [9] and IT-related applications and programs [10].

* Corresponding author. Tel.: +60-17-688-4094.
E-mail address: arafiei.sh@gmail.com.

Other important areas benefiting from HS include various medical studies, and their protocols [11, 12, 13] as well as improving and changing the reconfigurable mobile robot prototypes [14]. At the same time, research is also being carried out to improve the HS model itself since the past ten years. Various gurus of different fields have been able to mould the HS model to suit their own design and project needs. [15] could do just that for his project on rehabilitation of pipe networks. [16] employed a revised HS memory management approach and Pareto-dominance technique to handle the multi-modal and constrained optimization problems. For the improvement of designs in steel frames, [17] used the same technique and changed it to produce a harmony memory. It has also benefited by changes via the variable pitch adjustment rate or par [18]. [19] could show that when the fret width was adjusted for current populations, the results were more accurate and significant. However, he was sure to state that the fret adjusted can only be acceptable in cases where the *hmcr* or the harmony memory considering the rate was near to or equal to one.

The HS algorithm can be considered as universally acceptable, and has many advantages. It is different from other similar programs as it can utilize more than one search point at the same time. It is independent of the objective function derivative and can achieve optimum values of such objective functions, both at global or near-global optimization. It can take up high dimensional domains in this regard. Therefore, in many ways, the HS algorithm can be applied to both simple as well as complex analogies with good results and outcomes.

The advantages are met with one strong disadvantage. Typically, an HS algorithm, is computationally intensive and during its optimum finding process, requires longer and multiple objective function evaluation.

It was interesting to note that while there is much research on the HS algorithm, its applicability and its use, there was very little research carried out pertaining to the harmony memory updating. It is this lack of research that has led to the creation of this particular study and modification. This modification aims to reduce the number of improvisations that are needed in its processing in order to attain fast fitness achievement. This study will first evaluate the basic HS algorithm and how it works. This will be followed by the new innovations and modifications carried out on it. Finally, the two forms of HS algorithms will be compared, and two examples used to analyze the results from both types to assess the difference in outcomes.

2. Original HS method

The philosophy of the HS algorithm is essentially much different from other types of meta-heuristic optimization algorithms. This is because the other algorithms derive their inspiration from different natural processes. The HS algorithm, on the other hand, is inspired from the musical improvisation techniques. Just as the group of musician's search for a perfect harmony to create the best piece of music with the orchestra, the same is to find the best optimal solution for an optimization problem. According to [20], within a musical improvisation, there are three elements of correspondence carried out:

- The musical note is played exactly from musician’s memory. The same similarity is sought for memory usage in HS algorithm.

- In order to adjust the pitch and other musical harmony issues, a musician may attempt to experiment with slightly similar aforementioned notes in order to make the music notes sound better. This is similar to the pitch adjusting issues pertaining to HS algorithm.
- Putting in a random note or improvising of music within the piece, which is similar to the randomization carried out in HS algorithm.

The following steps are undertaken to apply HS algorithm by the given elements:

Step 1: The primary harmony memory is created, as shown below, where x_j^i is the j^{th} objective function variable in i^{th} randomly selected solution.

$$M = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_n^1 \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^s & x_2^s & \cdots & x_n^s \end{bmatrix} \qquad (1)$$

Equation (1) is formed through random variables, which can be produced in desired ranges.

Step 2: Substituting each row from the harmony memory matrix, defined by *M*, in objective function, $f(x_1, x_2, \cdots, x_n)$, *S* solutions are obtained. Here *F* represents the objective function solutions' matrix.

$$F = \begin{bmatrix} f(x_1^1, x_2^1, \ldots, x_n^1) \\ f(x_1^2, x_2^2, \ldots, x_n^2) \\ \vdots \\ f(x_1^s, x_2^s, \ldots, x_n^s) \end{bmatrix} \qquad (2)$$

Step 3: A new harmony is created from the harmony memory matrix. One of the rows of the matrix *M* is selected. This selection is based on the probability of harmony memory considering rate or *hmcr* mentioned previously. This is considered as the new potential solution candidate. A uniform random number is attained, called as *R*, which has the range of *[0,1]*. This is then compared with the *hmcr*. Should the *hmcr* be found greater than *R*, a new solution is then selected from the i^{th} randomly selected row of the *M*.

$$X^{new} = [x_1^{new} \quad x_2^{new} \quad \cdots \quad x_n^{new}] = [x_1^i \quad x_2^i \quad \cdots \quad x_n^i] \qquad (3)$$

By taking values close to it, and by using a probability of pitch adjusting rate or *par*, the new solution can be further muted. The new candidate is taken from the harmony memory with probability of *hmcr*, which is then subjected to be tuned by probability of *par*. In this way, when taking the case of P_{mute}, the mutation's total probability becomes.

$$P_{mute} = hmcr \times par \qquad (4)$$

As per the practice carried out, the pitch is usually adjusted linearly. This leads to the modification to the equation (3), via mutation, to find a new candidate solution. The following formula is then used.

$$X^{new} = [x_1^{new} \quad x_2^{new} \quad \cdots \quad x_n^{new}] = [x_1^i \quad x_2^i \quad \cdots \quad x_n^i] + [bw_1 \quad bw_2 \quad \cdots \quad bw_n] \times RND \qquad (5)$$

where $bw_j, (j = 1,2,\ldots,n)$, is pitch bandwidth, *RND* is a uniform random distribution number with the range of *[-1, 1]*.

***Step 4*:** In the case where the value of *hmcr* is found to be less than *R*, there is no need to carry out the third step. The new candidate solution, X^{new}, is taken from the entire set randomly. Its probability becomes *1-hmcr*. The process enables creating a wider domain of search within the search and helps in reaching global optimization.

Step 5: The next step is called updating of harmony memory. In this step, $f(X^{new})$ is is compared with the worst array of the matrix *F*, $f^{worst}(X^{worst})$, minimum array of *F* in minimization problem or maximum array of *F* in maximization problem. If $f(X^{new})$ is less than $f^{worst}(X^{worst})$ in minimization problem or is greater than $f^{worst}(X^{worst})$ in maximization problem, the new objective solution is replaced with the worst array. Consequently, X^{new} is also replaced by the X^{worst} in the matrix *M*.

Step 6: Steps 3 until 5 are repeated for the number of iterations that were assigned before hand is reached for the termination criterion.

3. Modified HS method

Our modified HS algorithm incorporates a new technique in order to update the harmony memory. The modification follows in the later stage at Step 5, where as other HS steps are similar to the basic method. The basic algorithm method operates in the following manner. When a candidate solution wins to place in harmony memory, it removes only one member of the harmony memory having the worst reputation. It then replaces it. The modified method is slightly different. Based on the Steps 3 or 4, the objective function of the new solution $f(X^{new})$, is computed and comparisons made with all arrays of the objective function matrix F in the equation (2).

Hence the comparison is drawn between $f(X^{new})$ and arrays having worse values against it. This is followed by replacement of the X^{new} with harmony memory members having objective functions lesser than the $f(X^{new})$ in minimization problems or have greater values in maximization problems. The new HS algorithm, therefore, removes members of harmony memory having lesser reputations, and replaces them the winner candidate solution. In other words, in the modified algorithm, we may have changes in many members of the *M* during each improvisation while in the original algorithm only one member of the *M* is updated in each improvisation.

4. Implementation and results

The following is four examples created to compare the outcomes of the basic HS with the new HS algorithm. This will help us in identifying the capability of the modified algorithm. The areas of comparison include the maximum number of improvisations, accuracy, and subsequent to it, fast performance convergence for four famous objective functions.

4.1. Rosenbrock's banana function

[21] introduced the Rosenbrock's valley or Rosenbrock's banana function. This function is used as a test problem to detect performance among optimization algorithms, and is non-convex in nature.

$$f(x,y) = \ln[1+(1-x)^2 + 100(y-x^2)^2] \qquad (6)$$

It is very difficult to achieve convergence to a global minimum. This is because the global minimum, defined as $f_{\min} = 0$, takes place in a long, narrow, parabolic shaped flat valley at *(1,1)*. This function was successfully used to prove the worth of the HS algorithm by [20]. In order to ensure that the comparisons were carried out equally, the random numbers generated from the HS modified algorithm version were placed simultaneously used in basic algorithm in the basic version as well. The same initial harmony memory is also used in both types of algorithms. This helped in pairing of the comparisons, thus clearly showing results from each set of comparison. The recommended parameter setting for both types of algorithms is shown in Tab. 1. The definition of each value corresponds with the values stated in the traditional values of the [20]. These include *S*, which is the number of harmony memory solution, *hmcr* which is the probability of the harmony memory considering the rate, the *par* which is the probability of pitch adjusting rate, and *bw* which is the bandwidth. The algorithm was set to stop at a certain predefined number of improvisations. In this manner, the optimal value of the Rosenbrock's banana function by the basic and the modified HS algorithm can be examined, and comparisons drawn on the two methods for the number of improvisations which the optimum value is achieved.

Table 1. HS parameters setting for Rosenbrock's banana function

S	***hmcr***	***par***	***bw***	**Search Domain of Variable *x***	**Search Domain of Variable *y***	**Pre-selected Number of Improvisations**
20	0.95	0.7	1	[-10, 10]	[-10, 10]	600

Tab. 2 illustrates the results obtained. It was found that the minimum value approximation for Rosenbrock's banana function was attained at very different iterations for each type of HS method. The traditional method showed it at 433 iterations, while 225 iterations were needed by the modified HS method. The achieved optimum value was more accurate for the modified HS when compared to the basic HS algorithm, despite using the same random numbers and parameters setting operations.

Table 2. Comparison of basic and modified HS

	x*	***y****	***f(x*, y*)***	**Number of Improvisations**
Basic *HS*	1.0140	1.0282	1.9598×10^{-4}	433
Modified *HS*	1.0059	1.0118	3.4930×10^{-5}	225

Fig. 1(a) and Fig. 1(b) both show the convergence history of the performance of both types of HS algorithm methods.

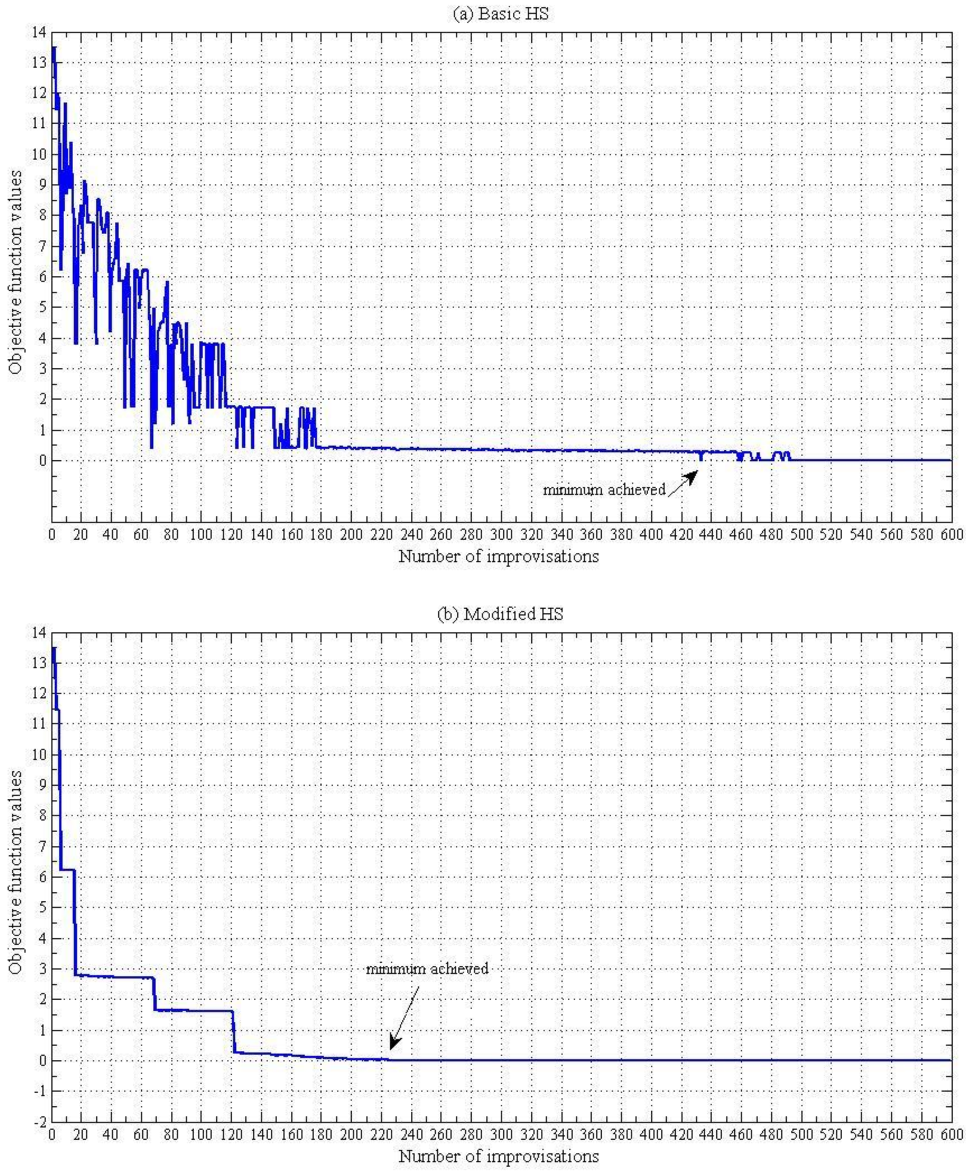

Fig. 1. (a) convergence of the minimization of the Rosenbrock's banana function by basic HS algorithm; (b) convergence of the minimization of the Rosenbrock's banana function by modified HS algorithm

As mentioned, in the HS modified method, if a candidate solution wins to place in harmony memory, it eliminates the members of the harmony memory having lesser reputations, and then replaces them. This helps in creating a descending trend throughout the convergence history in the modified HS version, as displayed in Fig. 1(b). The levelling off the graph after 225 improvisations is the indication that not any better optimum solution could be found. The same is not true to the traditional type of HS. In the traditional form, only one poor result is eliminated and replaced. This leads to the fluctuation in the convergence performance as shown in Fig. 1(a). The fluctuations in the graph remain even after the optimum value for the first time is achieved.

4.2. Michalewicz's Bivariate function

In the second example, the use of Michalewicz's Bivariate function [22] will take place. This is a multimodal test function that is carried out for optimization needs and can be seen below:

$$f(\vec{X}) = -\sum_{i=1}^{n} \sin(x_i)[\sin(\frac{i \times x_i^2}{\pi})]^{2m} \qquad (7)$$

where the parameter *m* defines the "steepness" of the valleys or edges and is assumed to be 10 for this solution. A global minimum happens at -1.801 for *n=2*. To draw same comparisons, the same random numbers were assigned to both types of HS analysis. The comparisons, therefore, are accurate in the sense that the same values are being used for evaluation. Recommended parameters adjusting for both, basic and modified HS algorithms, which are run concurrently with same initial harmony memory have been demonstrated in Tab. 3. The improvisations were set at 150 after which the algorithm would stop.

Table 3. HS parameters setting for Michalewicz's Bivariate function (n=2)

S	***hmcr***	***par***	***bw***	**Search Domain of Variable *x***	**Search Domain of Variable *y***	**Pre-selected Number of Improvisations**
20	0.9	0.3	1	$[0, \pi]$	$[0, \pi]$	150

The results of this method are seen in the Tab. 4. In case of the Michalewicz's Bivariate function, the minimum value for the original HS was attained at 62 iterations. For the modified HS method, the optimum value was found at 49 iterations. Again, the modified HS was superior to the basic one in terms of precise optimum value, when the same random numbers and parameter setting operations were used on both types of HS methods.

Table 4. Comparison of basic and modified HS

	x*	***y****	***f(x*, y*)***	**Number of Improvisations**
Basic *HS*	2.1830	1.5708	-1.7949	62
Modified *HS*	2.2029	1.5709	-1.8013	49

Fig. 2(a) and Fig. 2(b) show the convergence history of performance for both types of HS algorithm methods.

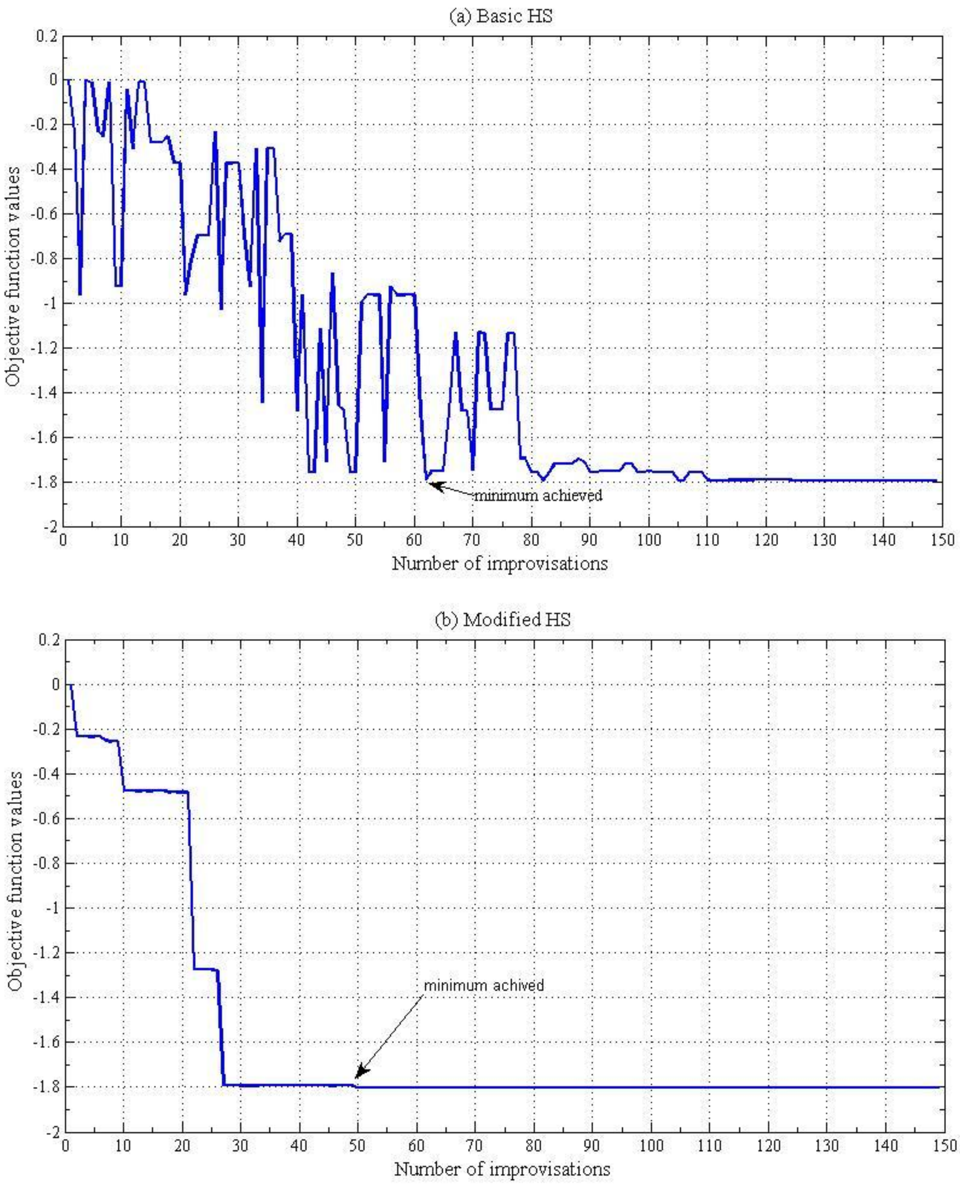

Fig. 2. (a) convergence of the minimization of the Michalewicz's Bivariate function by basic HS algorithm; (b) convergence of the minimization of the Michalewicz's Bivariate function by modified HS algorithm

Again, the similar findings to the Rosenbrock's banana function can be seen. The improved method displays a descending stepwise trend when viewing convergence history of performance. This was achieved because of the modified harmony memory update technique, shown in Fig. 2(b). Again, after 49 improvisations, the graph levels itself, showing that no more estimates can be found for better optimum solutions.

For *n=5* this function has a global optimum solution of -4.687. The solution space of this function has many local optimum solutions. Hence, the solution of this function using gradient-based optimization algorithms is quite difficult.

This function is also solved by both basic HS and modified HS algorithms for *n=5*. Recommended parameters adjusting for both, basic and modified HS algorithm have been demonstrated in Tab. 5.

Table 5. HS parameters setting for Michalewicz's Bivariate function (n=5)

S	***hmcr***	***par***	***bw***	**Search Domain of Variables** x_i ***(i=1…5)***	**Pre-selected Number of Improvisations**
30	0.9	0.5	1	[0, π]	8000

Tab. 6 shows the comparison of the identified results for each algorithm. As it can be seen from Tab. 6, although both basic and modified HS algorithms find near optimum solutions; the identified results of the modified HS algorithm more closely agree with the global optimum solution. Also, modified HS algorithm again requires much fewer iterations than basic HS algorithm.

Table 6. Comparison of basic and modified HS

	x_1*	x_2*	x_3*	x_4*	x_5*	$f(x_1^*, \ldots, x_5^*)$	**Number of Improvisations**
Basic *HS*	2.202852084	1.570787192	1.284957764	1.923045575	1.720492085	-4.687657338	2929
Modified *HS*	2.202905487	1.570796543	1.284992537	1.923057663	1.720468659	-4.687658088	1938

The comparison of the convergence histories for each case can be seen in Fig. 3. The result of empirical study indicates that the proposed HS algorithm can find better results than basic HS algorithm with much fewer iteration numbers.

4.3. Eason and Fenton's gear train inertia function

Eason and Fenton's function [23] is a minimization problem for the inertia of a gear train. The function is:

$$f(x_1, x_2) = 0.1(12 + x_1^2 + \frac{1 + x_2^2}{x_1^2} + \frac{x_1^2 x_2^2 + 100}{(x_1 x_2)^4}) \qquad (8)$$

The function has an optimum solution at (1.7435, 2.0297) with a corresponding function value of 1.74 while the bounds of decision variables are: $x_1 \leq 10$ and $x_2 \geq 0$.

Again, to draw same comparisons, the same random numbers were assigned to both types of HS algorithms. Recommended parameters adjusting for both, basic and modified HS algorithms, which are run concurrently with same initial harmony memory have been demonstrated in Tab. 7.

Table 7. HS parameters setting for Eason and Fenton's gear train inertia function

S	***hmcr***	***par***	***bw***	**Search Domain of Variable x1**	**Search Domain of Variable x2**	**Pre-selected Number of Improvisations**
50	0.9	0.5	1	[0, 10]	$[0, \infty)$	800

The comparison of the identified results for basic and modified algorithms is listed in Tab. 8. While basic HS algorithm finds the optimum solution after 411 improvisations, a better solution has been obtained after 213 improvisations using the modified HS algorithm. This is one of the most important advantages of the proposed algorithm which has been demonstrated in given examples.

Table 8. Comparison of basic and modified HS

	x_1^*	x_2^*	$f(x_1^*, x_2^*)$	**Number of Improvisations**
Basic *HS*	1.71810	2.07941	1.74476	411
Modified *HS*	1.76248	2.06761	1.74468	213

The comparison of the convergence histories for each algorithm can be seen in Fig. 4. The results of four empirical studies indicate that the proposed HS algorithm can obtain better results than basic HS algorithm with much fewer improvisation numbers.

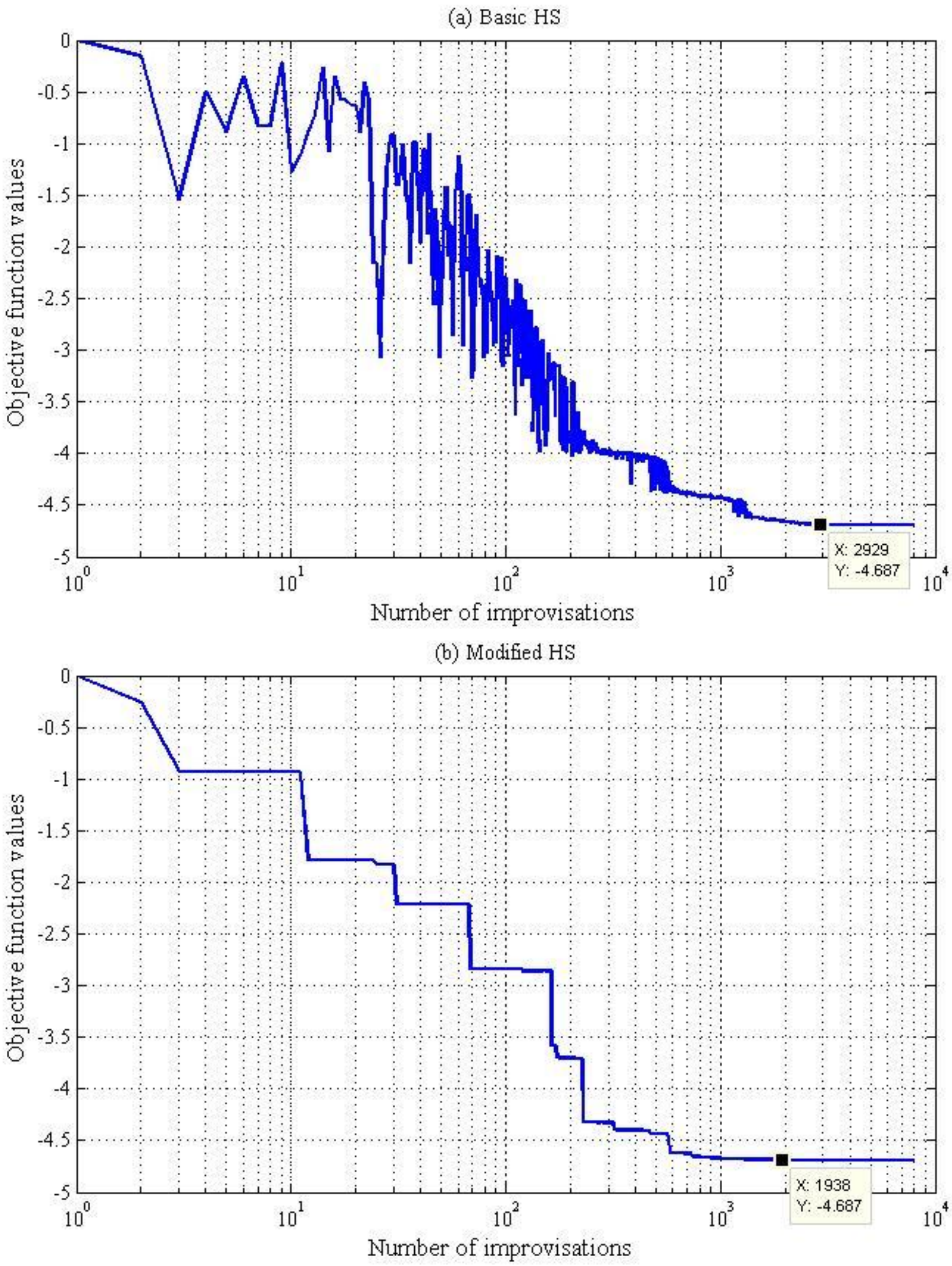

Fig. 3. (a) convergence of the minimization of the Michalewicz's Bivariate function by basic HS algorithm; (b) convergence of the minimization of the Michalewicz's Bivariate function by modified HS algorithm

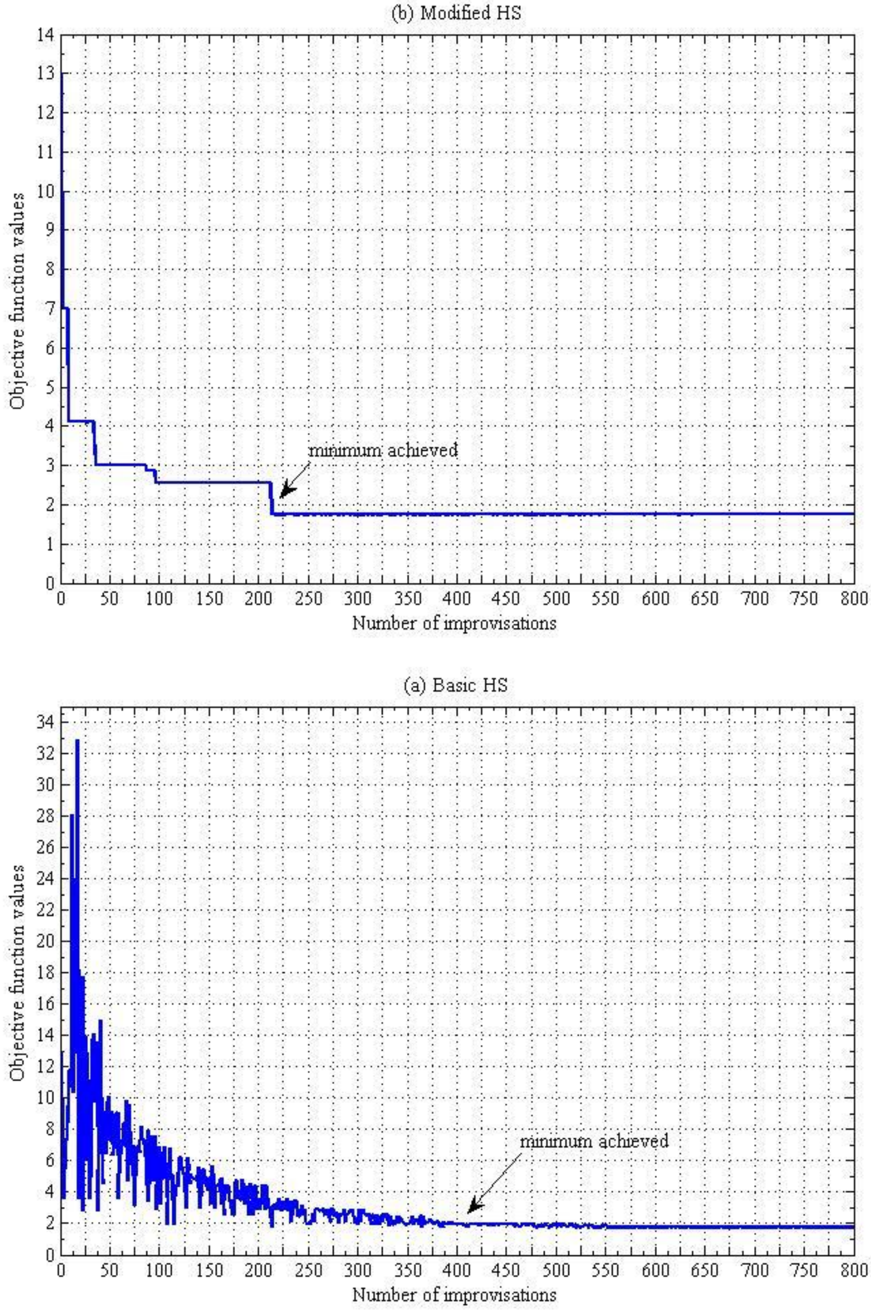

Fig. 4. (a) convergence of the minimization of the Eason and Fenton's gear train inertia function by basic HS algorithm; (b) convergence of the minimization of the Eason and Fenton's gear train inertia function by modified HS algorithm

5. Conclusions

The paper introduces a new version of the HS algorithm. This method has been found to reduce the maximum number of improvisations, for fast performance convergence. The algorithm was tested on four optimization problems, the Rosenbrock's banana function, the Michalewicz's Bivariate function for n=2 and n=5, and Eason and Fenton's gear train inertia function respectively. The coded algorithm used the same settings of operations, same paired comparisons, and the same random number generations to provide a fair comparison for basic and modified HS algorithms. Results showed there is a significant reduction in the maximum number of improvisations in the modified HS algorithm. Also, the new version of the updating technique obtained more accurate optimum solutions in far fewer number of improvisations compared with basic HS algorithm.

References

[1] Geem ZW. *Optimal design of water distribution networks using harmony search*. PhD thesis. South Korea: Korea University; 2000.

[2] Geem ZW, Kim JH, Loganathan GV. Harmony search optimization: application to pipe network design. *International Journal of Modelling and Simulation* 2002; **22**: 125-133.

[3] Geem ZW, Hwangbo H. Application of harmony search to multi-objective optimization for satellite heat pipe design. *Proceedings of US-Korea Conference on Science, Technology, and Entrepreneurship* 2006; Teaneck, NJ, USA.

[4] Geem ZW. Comparison harmony search with other meta-heuristics in water distribution network design. *8th Annual Water Distribution Systems Analysis Symposium* 2006; Cincinnati, USA.

[5] Lee KS, Geem ZW. A new structural optimization method based on the harmony search algorithm. *Computers & Structures* 2004; **82**: 781–798.

[6] Lee KS, Geem ZW, Lee SH, et al.. The harmony search heuristic algorithm for discrete structural optimization. *Engineering Optimization* 2005; **37**: 663–684.

[7] Lee KS, Choi CS. Discrete-continuous configuration optimization methods for structures using the harmony search algorithm. *Key Engineering Materials* 2006; 324-325, 1293–1296.

[8] Ayvaz MT. Simultaneous determination of aquifer parameters and zone structures with fuzzy c-means clustering and meta-heuristic harmony search algorithm. *Advances in Water Resources* 2007; **30**: 2326-2338.

[9] Geem ZW. Harmony search applications in industry. In *Soft Computing Applications in Industry* 2008; Berlin: Springer-Verlag; 117-134.

[10] Forsati R, Haghighat AT, Mahdavi M. Harmony search based algorithms for bandwidth-delay-constrained least-cost multicast routing. *Computer Communications* 2008; **31**: 2505-2519.

[11] Alexandre E, Cuadra L, Gil-Pita R. Sound Classification in Hearing Aids by the Harmony Search Algorithm. In *Music Inspired Harmony Search Algorithm* 2009; Geem ZW. Berlin: Springer; 173-188.

[12] Panchal A, Tom B. Harmony search optimization for HDR prostate brachytherapy. *International Journal of Radiation Oncology * Biology * Physics* 2009; **75**: S720-S721.

[13] Dong H, Bo Y, Gao M, Zhu T. Improved harmony search for detection with photon density wave. In *International Symposium on Photoelectronic Detection and Imaging: Related Technologies and Applications. (SPIE China)* 2007; 662523.

[14] Xu H, Gao XZ, Wang T. Harmony Search Optimization Algorithm: Application to a Reconfigurable Mobile Robot Prototype. In *Recent Advances in Harmony Search Algorithm* 2010; Geem ZW. Berlin: Springer; 11-22.

[15] Kim JH, Baek CW, Jo DJ, Kim ES, Park MJ. Optimal planning model for rehabilitation of water networks. *Water Science and Technology: Water Supply* 2004; **4**: 133-147.

[16] Gao XZ, Wang X, Ovaska SJ. Harmony Search Methods for Multi-modal and Constrained Optimization. In *Music Inspired Harmony Search Algorithm 2009*; Geem ZW. Berlin: Springer; 39-51.

[17] Degertekin S. Optimum design of steel frames using harmony search algorithm. *Structural and Multidisciplinary Optimization* 2008; **36**: 393-401.

[18] Mahdavi M, Fesanghary M, Damangir E. An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation* 2007; **188**: 1567-1579.

[19] Mukhopadhyay A, Roy A, Das S, Abraham A. Population-variance and explorative power of harmony search: an analysis. *Proceedings of 3rd IEEE International Conference on Digital Information Management* 2008; London: IEEE; 775-781.

[20] Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: Harmony search. *Simulation* 2001; **76**: 60-68.

[21] Rosenbrock HH . An automatic method for finding the greatest or least value of a function. *The Computer Journal* 1960; **3**: 175-184.

[22] Michalewicz Z. Genetic Algorithm + Data Structure = Evolution Programs. New York: Springer-Verlag; 1992.

[23] Eason ED, Fenton RG. A comparison of numerical optimization methods for engineering design. *ASME journal of engineering for industry* 1974; **96 (1)**: 196–200.