RECENT ADVANCES IN HARMONY SEARCH ALGORITHM

Assif Assad
Department of Computer Science and Engineering, Islamic University of Science and Technology Awantipora, India

ABSTRACT

Harmony Search (HS) a meta heuristic algorithm inspired by music improvisation process in which the musician searches for the best harmony and continues to polish the harmony in order to improve its aesthetics. The HS algorithm was introduced in the year 2001 and has found applications in diverse fields.

This manuscript reviews the recent significant developments in the structure of Harmony Search algorithm and also describes its recent state-of-the-art applications. As evidenced by a number of studies, this algorithm features several innovative aspects in its operational procedure that foster its utilization in diverse fields such as engineering, construction, telecommunications, robotics, health and energy

Keywords: Search, Evolutionary Algorithms, metaheuristic Optimization, Nature inspired Optimization.

Cite this Article: Assif Assad, Recent Advances in Harmony Search Algorithm, International Journal of Computer Engineering and Technology, 10(1), 2019, pp. 157-165.

http://www.iaeme.com/IJCET/issues.asp?JType=IJCET&VType=10&IType=1

1. INTRODUCTION

Optimization is defined as the process of selecting the best option from a set of available alternatives. Every process has the potential to be optimized and many challenging problems in business, economics, science and engineering can be formulated as an optimization process. For example, the objective of formulated optimization problems can be the maximization of profit and/or quality, or minimization of time, cost and risk etc.

The application of optimization algorithms to real world problems has gained momentum in the last decade. Dating back to the early 1940s, diverse traditional mathematical methods such as linear programming (LP), nonlinear programming (NLP) or dynamic programming (DP) were first employed for solving complex optimization problems by resorting to different relaxation methods of the underlying formulation. These techniques are capable of cost-efficiently obtaining a global optimal solution in problem models subject to certain particularities (e.g. optimal sub structurability and subproblem overlap for dynamic programming), but
unfortunately their application range does not cover the whole class of NP-complete problems, where an exact solution cannot be found in polynomial time. In fact, the solution space (and hence, the solving time) of the problem increases exponentially with the number of inputs, which makes them unfeasible for practical applications.

Many real life optimization problems are complex and thus difficult to solve in an exact manner within reasonable amount of time. The classical optimization methods have the limitation of being highly sensitive to the starting point and frequently converge to a local optimum. Classical optimization techniques can be broadly classified into two categories: direct search method and gradient based search method. In the direct search method, only the objective function and constraints are utilized for the search process, whereas in the gradient based search method, the first order and/or second order derivatives are utilized in the search process. Direct search methods have the limitation of slow convergence whereas gradient based search methods are faster however have the limitation of being not applicable to discontinuous and non differentiable functions. Furthermore, both methods seek local optima, thus starting the search in the vicinity of a local optima causes them to miss the global optima.

Metaheuristic algorithms eradicate some of the aforementioned difficulties and are quickly replacing the classical methods in solving complex non linear optimization problems. Metaheuristic algorithms typically intend to find a reasonably good solution close to optimization in reasonable amount of computational time. During the last few decades, several metaheuristic algorithms have been proposed including Genetic Algorithms, Genetic Programming, Evolutionary Programming, Evolutionary Strategies (ES), Differential Evolution, Ant Colony Optimization and Particle Swarm Optimization to name a few.

The Harmony Search (HS) algorithm is the musicians inspired metaheuristic algorithm developed by Geem et.al. in 2001 [1] and has found applications in diverse fields. Weyland [2] raised an issue regarding the novelty of Harmony Search algorithm by declaring it a special case of ES, however the pitch adjustment operator used in HS is entirely different than the mutation operator used in ES further HS utilizes the pitch adjustment operator (local search) probabilistically determined by PAR and thus the two cannot be considered same. Ample evidence has been provided by Saka et.al. in [3] to show HS is not a special case of ES even though superficially they seem to be identical.

2. HARMONY SEARCH AND ITS VARIANTS

In this section an introduction to Harmony Search algorithm and some of its state-of-the-art variants is provided.

2.1. Harmony Search

Harmony Search is a musicians behavior inspired evolutionary algorithm developed in 2001 by Geem et al.[1], though it is a relatively new meta heuristic algorithm, its effectiveness and advantages have been demonstrated in various applications like design of municipal water distribution networks [4], structural design [5], traffic routing [6], load dispatch problem in electrical engineering [7], multi objective optimization [8], rostering problems [9], clustering [10], classification and feature selection [11, 12] to name a few. A detailed survey on applications of HS can be found in [13, 14]. In order to explain the Harmony Search in detail, let us first idealize the improvisation process by a skilled musician. When a musician is improvising there are three possible choices:

1. Play any piece of music exactly from his memory.
2. Play something similar to a known piece.
3. Compose new or random notes.
Geem et al. [1] formalized these three options into quantitative optimization process and the three corresponding components become usage of harmony memory (HM), pitch adjusting, and randomization. The usage of HM is similar to the choice of the best fit individuals in genetic algorithms. In order to use this memory effectively, it is typically assigned a parameter called harmony memory considering rate (HMCR 2 [0, 1]). If this rate is low (near 0), only few best harmonies are utilized and thus convergence of algorithm is slow. If this rate is very high (near 1), it results in exploitation of the harmonies in the HM, thus the solution space is not explored properly leading to potentially inefficient solutions. Typically HMCR 2 [.7, .95] is used. The second component is pitch adjustment determined by a pitch bandwidth (BW) (also referred as fret width [15] ) and a pitch adjusting rate (PAR), it corresponds to generating a slightly different solution in the HS algorithm. Pitch can be adjusted linearly or nonlinearly however most often linear adjustment is used. So we have

\[ H_i^{\text{new}} = H_i^{\text{old}} + \text{BW} \times r_i \quad \text{where } r_i \in [1, 1] \quad \text{and } 1 \leq i \leq D \quad (1) \]

Where \( H_i^{\text{old}} \) is the \( i^{\text{th}} \) component of the existing harmony or solution and \( H_i^{\text{new}} \) is the \( i^{\text{th}} \) component of new harmony after the pitch adjusting action and BW is the bandwidth. The Equation (1) essentially produces a new solution around the existing solution by altering it slightly by a very small random amount. Here \( r_i \) is a random number generated in the range of [-1, 1] and D is total number of components in the harmony. The pitch adjusting rate (PAR) controls the degree of adjustment. A low pitch adjusting rate with a narrow bandwidth can slow down the convergence of HS because of limitation in exploration of only a small subspace of the whole search space. On the other extreme a very high PAR with a wide bandwidth may cause the algorithm to swing around some optimal solution. Thus the recommended value of PAR 2 [.1, .5]. The third component of the HS is the randomization, which is used to increase the exploration of the search space. Although pitch adjustment plays a somewhat similar role, but it is confined to close neighborhood of harmony and thus corresponds to local search. The use of randomization pushes the algorithm further to explore diverse search areas to find the global optima. The pseudo code of harmony search is shown as Algorithm 1. In the pseudo code H represents a potential solution or harmony, rand [0, 1] is a uniformly distributed random number generator, rand_int(1,HMS) generates a uniformly distributed integer random number between 1 and HMS, size of harmony memory is represented as HMS and D is the dimension of problem.

Algorithm 1 HARMONY SEARCH (HS) ALGORITHM

- Define Objective function \( f(H) \)
- Define harmony memory consideration rate (HMCR).
- Define pitch adjustment rate (PAR) and bandwidth (BW).
- Define Harmony Memory Size (HMS).
- Initialize Harmony Memory (HM).
- while (StoppingCriteria Not Reached) do
  - Find current Worst and Best harmony in HM.
  - for \( i = 1 \) to \( D \) do
    - if (rand \( \leq \) HMCR) then
      - \( H_i = H_{M/w} \) where \( w = \text{rand\_int(1,HMS)} \)
    - if (rand \( \leq \) PAR) then
      - \( H_i = H_i + \text{ rand \_BW} \)
    - end if
  - else
    - Generate \( H_i \) randomly within the allowed bounds.
  - end if
  - end for
- if (H is better than worst Harmony in HM) then
  - Update HM by replacing WORST harmony by H.
- end if
- end while
- print Best Harmony as obtained solution.
2.2. Harmony Search variants based on handling of parameter

In order to enhance the performance of the standard HS algorithm several variants of HS algorithm has been proposed in literature. The parameters- HMCR, PAR and BW remain constant in standard HS. In order to strike a balance between exploration and exploitation the parameters of HS are dynamically altered in the algorithm giving rise to different variations of the HS algorithm. A detailed survey on variants of HS can be found in [16].

2.2.1. Improved Harmony Search

Mahdavi et.al. [17] proposed dynamic adaptation of both pitch adjustment rate (PAR) and bandwidth (BW), the algorithm became known as Improved Harmony Search (IHS). In Improved Harmony Search PAR is linearly increased in each iteration using Equation (2) and BW is exponentially decreased in each iteration using Equation (3).

\[
PAR_{gn} = PAR_{min} + \frac{(PAR_{max} - PAR_{min})}{NI - 1} \times (gn - 1) \tag{2}
\]

where \(PAR_{gn}\) is pitch adjusting rate for each generation, \(PAR_{min}\) is the minimum pitch adjusting rate, \(PAR_{max}\) is the maximum pitch adjusting rate, \(NI\) is the Maximum number of generations, \(gn\) is generation number.

\[
BW_{gn} = BW_{max} \times \exp(c \times (gn - 1)) \text{ where } c = \frac{\ln(BW_{min} / BW_{max})}{NI - 1} \tag{3}
\]

In Equation (3) \(BW_{gn}\) represents bandwidth for each generation, \(BW_{min}\) and \(BW_{max}\) is the minimum and maximum bandwidth respectively. All the other steps in IHS are similar to that of standard Harmony Search algorithm.

2.2.2. Global Best Harmony Search

Inspired from Particle Swarm Optimization paradigm Omran and Mahdavi introduced another important modification to Harmony Search algorithm referred as Global-best Harmony Search (GHS) [18] algorithm. In GHS the parameter PAR is linearly increased as in IHS algorithm, however the concept of BW has completely been removed and thus pitch adjustment step (Step 12 of Algorithm 1) has been modified as Equation (4).

In order to generate \(i^{th}\) component of the harmony \(H\), in pitch adjustment step \(k^{th}\) component of the best harmony in HM is assigned to it.

\[
H_i = HM_k^{Best} \text{ where } i, k \in \{1, 2, ..., D\} \text{ and } D \text{ is the dimension of harmony.} \tag{4}
\]

Assigning the \(k^{th}\) component of the best harmony to the \(i^{th}\) component of the new harmony in Equation (4) is contentious because most of the times different components of problem dimension are independent of each other and there may be vast differences in search ranges from one dimension to another. Further directing the search towards the best harmony in HM causes a serious side effect of premature convergence due to lack of diversity in HM.

2.2.3. Adaptive Harmony Search algorithm

A new adaptation for HS was proposed in [19, 20] by changing HMCR and PAR dynamically during the execution of Harmony Search algorithm. Initially HMCR and PAR are respectively
set to $H_{MCR}^{(0)}$ and $PAR^{(0)}$ then the dynamic calculating of these parameters is adapted as follows:

$$H_{MCR}^k = (1 + \frac{1 - H_{MCR}'^k}{H_{MCR}'}) \times e^{-\gamma N(0,1)} - 1$$

(5)

$$PAR^k = (1 + \frac{1 - PAR'^k}{PAR'^}) \times e^{-\gamma N(0,1)} - 1$$

(6)

where $H_{MCR}^k$ and $PAR^k$ are the sampled values of the adapted parameters for a new harmony vector. $N(0, 1)$ is a normally distributed random number in the range of 0 to 1, is called learning rate recommended to be in range of [0.25, 0.50]. $H_{MCR}^0$ and $PAR^0$ are the average values of improvisation parameters obtained by averaging the corresponding values of all the solution vectors within the HM matrix i.e.

$$H_{MCR}' = \frac{\sum_{i=1}^{HMS} H_{MCR}^i}{HMS}$$

(7)

$$PAR' = \frac{\sum_{i=1}^{HMS} PAR^i}{HMS}$$

(8)

The Adaptive Harmony Search algorithm has been utilized in problems like structural optimization and designing code optimization of steel structures however it has not been tested on standard benchmark functions.

Self-adaptive Harmony Search was proposed by Wang et.al. in [21]. The authors of Self adaptive Harmony Search introduced three modifications in standard HS. In the initialization step a low discrepancy sequence is used to have more uniformly initialized harmonies in the HM. The second important alteration is to decrease PAR during execution rather than increasing it, to prevent overshooting and oscillation around the optimal solution. The third modification

Is the removal of BW parameter and changing the pitch adjustment step as shown below.

$$trial^i = (max(HM^i) - trial^i) \times ran[0, 1]$$

(9)

$$trial^i = (trial^i - min(HM^i)) \times ran[0, 1]$$

(10)

where $ran[0, 1)$ is a uniformly distributed random number ranged from 0 to less than 1, $trial^i$ is the $i^{th}$ variable selected from HM, and $max(HM^i)$ and $min(HM^i)$ are respectively highest and lowest values of the $i^{th}$ variable in the HM. In each step one of the Equations (9) or (10) is randomly selected to adjust the current pitch.

The authors of Self-adaptive Harmony Search have compared the algorithm with standard HS and GHS but on only four benchmark functions namely: Sphere, Rosenbrock, Ackley and Griewank. Further Self-adaptive Harmony Search has not been tested on hybrid and composite functions (as provided in IEEE CEC 2014 benchmark suite) which portray many real world problems.
2.2.5. Self-adaptive Global Best Harmony Search

Pan et al. modified the GHS algorithm to create Self-adaptive Global Best Harmony Search (SAGHS) algorithm. The main difference between GHS and SAGHS is that pitch adjustment step (Step 12 of Algorithm 1) has been modified as Equation (11). Thus to generate \(i^{th}\) component of the harmony \(H\), in the pitch adjustment step \(i^{th}\) component of the best harmony in HM is assigned to it.

\[
H_i = H_{M_i}^{\text{Best}} \quad \text{where} \ i \in \{1,...,D\} \text{ is the dimension of harmony and } \text{Best is the index of best harmony in HM:}
\]

(11)

In order to increase diversity in HM, the harmony memory consideration step (Step 10 of Algorithm 1) is modified as Equation (12). Further the BW is dynamically updated in the algorithm as Equation (13).

In Equation (13) \(BW_{gn}\) represents bandwidth for each generation, \(BW_{\text{min}}\) and \(BW_{\text{max}}\) is the minimum and maximum bandwidth respectively.

\[
H_i = HM_i^j \pm \text{rand} \times BW \quad \text{where} \ i \in \{1,2,...,D\}, \ j \in \{1,2,...,\text{HMS}\}
\]

\(D\) is the dimension of harmony and \(\text{HMS}\) is the size of \(HM\).

(12)

\[
BW_{gn} = \begin{cases} 
BW_{\text{max}} - \frac{BW_{\text{max}} - BW_{\text{min}}}{N_I} \times 2gn & \text{if } gn < \frac{N_I}{2} \\
BW_{\text{min}} & \text{if } gn \geq \frac{N_I}{2}
\end{cases}
\]

(13)

Self-adaptive Global Best Harmony Search algorithm has been compared with standard HS, IHS and GHS algorithms on 16 benchmark functions however it has not been tested on hybrid and composite functions.

2.2.6. Intelligent Tuned Harmony Search Algorithm

The main idea in Intelligent Tuned Harmony Search (ITHS) algorithm [23] is to divide the harmony memory into two groups namely Group A and Group B. Group A is responsible for both exploration and exploitation and consists of all the harmonies whose objective function value is less than or equal to \(HM^{\text{mean}}\). Group B consists of the harmonies whose objective function value is greater than \(HM^{\text{mean}}\) and is responsible for exploration only. \(HM^{\text{mean}}\) is the mean of objective function value of all the harmonies in the HM. Further the value of PAR is dynamically updates as shown in Equation (14).

\[
PAR_{gn} = PAR_{\text{max}} + \frac{(PAR_{\text{max}} - PAR_{\text{min}})}{N_I} \times gn
\]

(14)

where \(PAR_{gn}\) is pitch adjusting rate for each generation, \(PAR_{\text{min}}\) and \(PAR_{\text{max}}\) is the minimum and maximum pitch adjusting rate respectively, \(N_I\) is the maximum number of generations, \(gn\) is generation number.

2.2.7. Improved Global-best Harmony Search

The Improved Global-best harmony search (IGHS) introduced in [24] has been created by modifying pitch adjust-ment step of the standard HS. In the pitch selection step a randomly selected pitch is adjusted according to Gaussian distribution as shown in equation below:

\[
H_i = HM_i^j + \text{Gauss}(0, 1) \times BW
\]

(15)
where Gauss(0,1) is a Gaussian random number with mean 0 and a standard deviation 1 and \( r \in \{1,2,...,\text{HMS}\} \). For the pitch adjustment step the current pitch of the best harmony is selected and adjusted using uniform distribution as shown in equation below:

\[
H_i = HM_i^{\text{Best}} + \text{rand}(0, 1) \times BW
\]  

(16)

where \( \text{rand}(0,1) \) is a uniformly distributed random number in the range [-1,1] and \( \text{Best} \) is the index of best harmony in HM.

The idea is to continuous perturb around the best harmony to improve the quality of the best solution obtained. The Gaussian distribution in Equation (15) provides a wide area of search around a random harmony and uniform distribution in Equation (16) provides a small area of search around the best harmony.

2.2.8. Other variants of Harmony Search based on handling of parameters

Cheng et al. [25] developed Modified Harmony Search (MHS), which is based on the idea of selecting better harmony with higher probability. Pan et al. proposed a local best Harmony Search algorithm with dynamic sub populations (DLHS) in [26]. In DLHS the HM is divided into many small sized sub HMs and then independent evolution is performed on each sub-HM. The sub HMs are regrouped frequently to exchange information and maintain diversity. Explorative Harmony Search (EHS) [27] algorithm proposed by Das et al. eliminates the limitation of tuning the BW parameter by making it proportional to the population variance in HM. Another variant of HS based on the idea of increasing the PAR rather than decreasing it so as to favor exploration in the beginning of the algorithm has been proposed in [28]. A Two phase Harmony Search has been proposed in [29].

3. CONCLUSION

This article introduced the recent significant developments in the structure of Harmony Search algorithm and also describes its recent state-of-the-art applications. As evidenced by a number of studies, this algorithm features several innovative aspects in its operational procedure that foster its utilization in diverse fields such as engineering, construction, telecommunications, robotics, health and energy.

REFERENCES

[1] Z. W. Geem, J. H. Kim, and G. Loganathan, “A new heuristic optimization algorithm: harmony search,” Simulation, vol. 76, no. 2, pp. 60–68, 2001.

[2] D. Weyland, “A rigorous analysis of the harmony search algorithm: how the research community can be,” Modeling, Analysis, and Applications in Metaheuristic Computing: Advancements and Trends: Advancements and Trends, p. 72, 2012.

[3] M. Saka, O. Hasanc¸ebi, and Z. Geem, “Metaheuristics in structural optimization and discussions on harmony search algorithm,” Swarm and Evolutionary Computation, vol. 28, pp. 88–97, 2016.

[4] Z. W. Geem, “Optimal cost design of water distribution networks using harmony search,” Engineering Optimization, vol. 38, no. 03, pp. 259–277, 2006.

[5] S. Gholizadeh and A. Barzegar, “Shape optimization of structures for frequency constraints by sequential harmony search algorithm,” Engineering Optimization, vol. 45, no. 6, pp. 627–646, 2013.

[6] 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, 1552–1557, 2005.
Recent Advances in Harmony Search Algorithm

[7] L. Wang and L.-p. Li, “An effective differential harmony search algorithm for the solving non-convex economic load dispatch problems,” International Journal of Electrical Power & Energy Systems, vol. 44, no. 1, pp. 832–843, 2013.

[8] K. Nekooei, M. M. Farsangi, H. Nezamabadi-Pour, and K. Y. Lee, “An improved multi-objective harmony search for optimal placement of dgs in distribution systems,” Smart Grid, IEEE Transactions on, vol. 4, no. 1, pp. 557–567, 2013.

[9] M. Hadwan, M. Ayob, N. R. Sabar, and R. Qu, “A harmony search algorithm for nurse rostering problems,” Information Sciences, vol. 233, 126–140, 2013.

[10] D. C. Hoang, P. Yadav, R. Kumar, and S. K. Panda, “Real-time implementation of a harmony search algorithm-based clustering protocol for energy-efficient wireless sensor networks,” Industrial Informatics, IEEE Transactions on, vol. 10, no. 1, pp. 774–783, 2014.

[11] R. Diao and Q. Shen, “Feature selection with harmony search,” Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 42, no. 6, pp. 1509–1523, 2012.

[12] H. Fattahi, A. Gholami, M. S. Amiribakhtiar, and S. Moradi, “Estimation of asphaltene precipitation from titration data: a hybrid support vector regression with harmony search,” Neural Computing and Applications, vol. 26, no. 4, pp. 789–798, 2015.

[13] D. Manjarres, I. Landa-Torres, S. Gil-Lopez, J. Del Ser, M. N. Bilbao, S. Salcedo-Sanz, and Z. W. Geem, “A survey on applications of the harmony search algorithm,” Engineering Applications of Artificial Intelligence, vol. 26, no. 8, pp. 1818–1831, 2013.

[14] A. Assad and K. Deep, “Applications of harmony search algorithm in data mining: A survey,” in Proceedings of Fifth International Conference on Soft Computing for Problem Solving, pp. 863–874, Springer, 2016.

[15] Z. W. Geem and Y. Yoon, “Harmony search optimization of renewable energy charging with energy storage system,” International Journal of Electrical Power & Energy Systems, vol. 86, pp. 120–126, 2017.

[16] O. Mohd Alia and R. Mandava, “The variants of the harmony search algorithm: an overview,” Artificial Intelligence Review, vol. 36, no. 1, pp. 1,49–68, 2011.

[17] 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.

[18] M. G. Omran and M. Mahdavi, “Global-best harmony search,” Applied Mathematics and Computation, vol. 198, no. 2, pp. 643–656, 2008.

[19] O. Hasanc,e, F. Erdal, and M. P. Saka, “Adaptive harmony search method for structural optimization,” Journal of Structural Engineering, vol. 136, no. 4, pp. 419–431, 2009.

[20] P. Saka, M and O. Hasanc,e, “Adaptive harmony search algorithm for design code optimization of steel structures,” in Harmony Search Algorithms for Structural Design Optimization, pp. 79–120, Springer, 2009.

[21] C.-M. Wang and Y.-F. Huang, “Self-adaptive harmony search algorithm for optimization,” Expert Systems with Applications, vol. 37, no. 4, 2826–2837, 2010.

[22] Q.-K. Pan, P. N. Suganthan, M. F. Tasgetiren, and J. J. Liang, “A self-adaptive global best harmony search algorithm for continuous optimization problems,” Applied Mathematics and Computation, vol. 216, no. 3, pp. 830–848, 2010.

[23] P. Yadav, R. Kumar, S. K. Panda, and C. Chang, “An intelligent tuned harmony search algorithm for optimisation,” Information Sciences, vol. 196, pp. 47–72, 2012.

[24] M. El-Abd, “An improved global-best harmony search algorithm,” Applied Mathematics and Computation, vol. 222, pp. 94–106, 2013.
Y. Cheng, L. Li, T. Lansivaara, S. Chi, and Y. Sun, “An improved harmony search minimization algorithm using different slip surface generation methods for slope stability analysis,” Engineering Optimization, vol. 40, no. 2, pp. 95–115, 2008.

Q.-K. Pan, P. Suganthan, J. Liang, and M. F. Tasgetiren, “A local-best harmony search algorithm with dynamic subpopulations,” Engineering Optimization, vol. 42, no. 2, pp. 101–117, 2010.

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, 89–106, 2011.

N. Taherinejad, “Highly reliable harmony search algorithm,” in Circuit Theory and Design, 2009. ECCTD 2009. European Conference on, 818–822, IEEE, 2009.

A. Assad and K. Deep, “A two-phase harmony search algorithm for continuous optimization,” Computational Intelligence.