Research Journal of Applied Sciences 8 (3): 191-198, 2013
ISSN: 1815-932X

The Harmony Search Algorithms in Solving Combinatorial Optimization Problems

Masri Ayob, Mohammed Hadwan and Hafiz Mohd Sarim
Data Mining and Optimisation Research Group (DMO), Centre for Artificial Intelligent (CAIT),
Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia

Abstract: Harmony Search Algorithm (HSA) is a relatively new nature-inspired algorithm. HSA is a population based metaheuristic that inspired by the improvisation of jazz musicians. It evolves solutions by mimicking the musical improvisation process in searching for agreeable harmony. Many variants of HSA have proposed to solve different types of optimization problems. They vary from hybridizing some concepts of other metaheuristic approaches to enhance the HSA to taking some components of HSA and employed them to enhance the other metaheuristic approaches. This research reviews research issues of the parameter settings in HSA and application of HSA to effectively solve some complex problems.

Key words: Harmony search, meta-heuristic, timetabling and personnel scheduling, HSA nature, inspired algorithm, Malaysia

INTRODUCTION

Harmony Search Algorithm (HSA) is a metaheuristic algorithm that inspired by the improvisation of jazz musicians by Geem *et al*. (2001). HSA is classified under the population-based metaheuristic algorithm embedded with local-based heuristic (Lee and Geem, 2004). In music performance each musical player plays one musical note each time. Those musical notes combined together to form a harmony. In optimization each variable during the optimization process has a value at each time those values all together form a solution vector (Ingram and Zhang, 2009).

Since, the first introduction of HSA fundamentals by Geem *et al*. (2001) many variants of this algorithm have appeared to solve different types of optimization problems. These variants vary from hybridizing some concepts of other metaheuristic approaches to the framework of HSA to taking some concepts of HSA and using them in the framework of other metaheuristic approaches.

HSA has its own special features that make it different from other metaheuristics. These features can be listed as follows: HSA explores the search space iteratively by combining different parts of the solution candidates from different solution regions to form a new solution. This makes the HSA utilize the harmony memory more efficiently than Genetic Algorithm (GA) which combines only two parents of solution vectors to form the new solutions (Geem, 2006). This feature makes HSA free from the building block theory that highly affects the mechanism of GA. HSA can independently consider each component variable when it generates a new vector. Whereas, genetic structure by moving from the current solution to its neighbors using pitch adjusting rate. HSA combines different characteristics of existing metaheuristic algorithms (Lee and Geem, 2004) by using the harmony memory, HSA is similar to tabu search algorithm (Glover and Laguna, 1993) in the ability of preserving the history of earlier vectors. Further, it is similar to genetic algorithm (Holland, 1975) in the ability of managing several solution vectors simultaneously. The dynamic change of the parameters of HSA during the run, acts it as simulated annealing algorithm (Kirkpatrick *et al*., 1983).

A good reference for the applications and developments of HSA is Ingram and Zhang (2009). Their applications of HSA included continuous and discrete optimization problems such as: vehicle routing, bus scheduling, examination timetabling, job shop scheduling, Sudoku puzzle solving, structural design, ground water modeling, image segmentation, medical image, energy system dispatch and water network design (Alia and Mandava, 2011; Ingram and Zhang, 2009).

The improvement of the basic HSA has attracted many researchers. Similar to other metaheuristics, basic HSA needs proper parameter setting in order to improve the search efficiency. Slow convergence issue can occur due to improper parameter setting. Those settings need to be carefully assigned depending on the problem in hand.

Corresponding Author: Masri Ayob, Data Mining and Optimisation Research Group (DMO), Centre for Artificial Intelligent (CAIT), Universiti Kebangsaan Malaysia, 43600 UKM Bangi Selangor, Malaysia

Therefore, the empirical experiments trials seem to be the only guidance to select the best parameter values. In this study, researchers review the basic HSA and the most important modifications and hybridizations found in literature of HSA.

BASIC HARMONY SEARCH ALGORITHM

Similar to a group of musicians where they develop their harmonies based on practice by practice; HSA improves a solution vectors based on iteration by iteration using a good candidate solution that have been discovered during construction of initial population (Geem *et al*., 2001). HSA uses stochastic random search that depends on the Harmony Memory Considering Rate (HMCR) and Pitch Adjusting Rate (PAR).

Analogy between improvisation and optimization: Before explaining the fundamentals of HSA, it is worth to see the relation between improvisation and optimization shown in Fig. 1 as follows:

- Each musician corresponds to each decision variable
- Musical instrument's pitch range corresponds to decision variable's value range
- Musical harmony at certain time corresponds to solution vector at certain iteration
- Audience's aesthetics corresponds to objective function
- Improving musical harmony practice after practice corresponds to improving the solution vector iteration by iteration

The basic structure of HSA: The procedure of basic HSA consists of five steps to be applied in order to do the improving process. These steps are:

- Initialize HSA parameters
- Build the Harmony Memory (HM)
- Improvise new solution

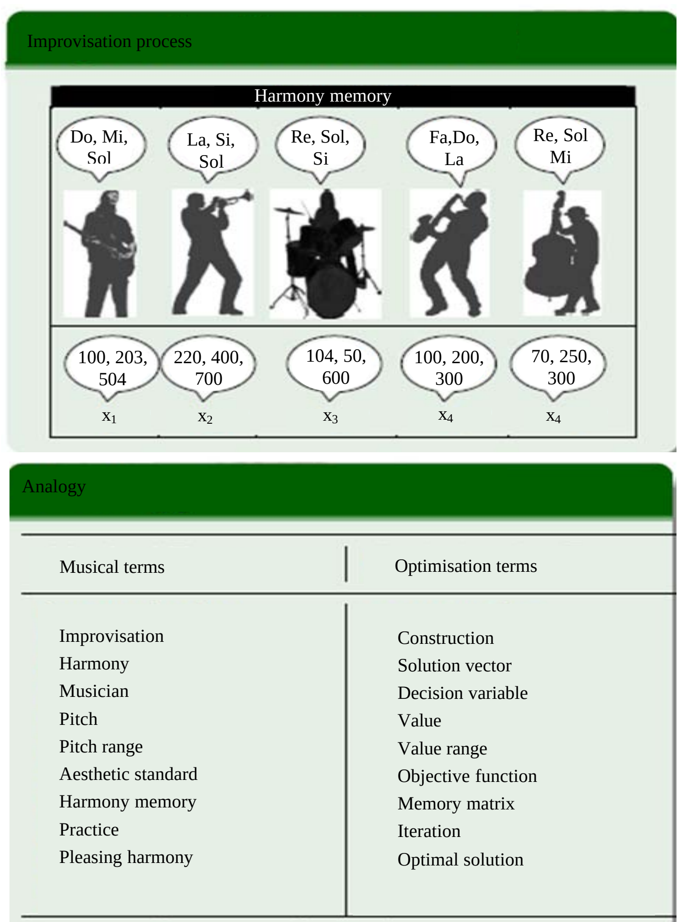

Fig. 1: The analogy between improvisation and optimization (Geem, 2010)

- Update HM
- Check the termination criteria

Let HMS as the harmony memory size, HMCR as the Harmony Memory Considering Rate, PAR as the Pitch Adjustment Rate, BW as the distance Bandwidth, NI as the Number of Improvisations (iterations), PS as a Population of Solution and p_i the index of the solution variables. Figure 2 shows a pseudo-code for basic HSA.

Initialize HSA parameters: In step 1, HSA parameters are initialized by setting up the values of the following parameters:

- Harmony Memory Size (HMS) represents the number of individual solution vectors that are sorted in the HM. HMS is similar to the total number of individuals in the population matrix of the GA
- Harmony Memory Considering Rate (HMCR) which varies between ($0 \leq HMCR \leq 1$) where HSA picks the candidate solution from HM. (1-HMCR) is the rate where the candidate solution is randomly picked from the entire PS. HMCR is similar to the crossover rate in GA
- Pitch Adjusting Rate (PAR) which varies between ($0 \leq PAR \leq 1$) where HSA change the value of theselected candidate solution from HM. The amount of changes is determined based on the value of BandWidth (BW). (1-PAR) is the rate where HSA keeps the value of the selected candidate solution the same without change
- The maximum Number of Improvisation (NI) to stop the search
- Optimization problem instance

Building the Harmony Memory (HM): In step 2, the HM matrix as shown in Eq. 1 is constructed by filling it up with

```
The Basic Harmony Search Algorithm (HSA)
begin
Initialize the parameters of HSA and optimization problem
    1: set the HSA parameters (HMS, HMCR, PAR, BW and NI)
    2: set the input instance of optimization problem
Initialize the Harmony Memory (HM)
    for (i = 1 to HMS) do
        xi = ø;
          for j = (1 to vector length) do
            choose p_i randomly from the PS
            add p_i to x_i
          Endfor
          calculate the penalty of x_i
          add x_i to HM
          sort the solutions based on its objective function value f (x)
     Endfor
 Improvise a new harmony (generate new solution)
        x^new = ø
        for i = (1 to NI) do
          if (rand(0,1) ≤HMCR) then
              choose variable p_i randomly from HM
                     if (rand(0,1) ≤ PAR) then
                  pitch adjusted xnew [p_i]
                    else
                  x^new[p_i] = x[p_i]
                endif
          else
              choose variable p_i randomly from the PS
                x^new [i] = p_i
          Endif
        Endfor
Update the HM
        if x^new is better than HM[HMS-1] then
           remove HM[HMS-1]
           add x^new to HM
           sort HM ascendingly
       Endif
Check the stopping criterion
    repeat step 3 and step 4 until reach the stopping criterion NI
    output the best harmony found so far
end
```

Fig. 2: The pseudo-code of basic HSA

randomly generated solutions. The number of solution vectors is equal to HMS that initialized in step1. Then, the HM is sorted based on the objective function (minimizing or maximizing):

$$HM = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_{i-1}^1 & x_i^1 \\ x_1^2 & x_2^2 & \cdots & x_{i-1}^2 & x_i^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{HMS-1} & x_2^{HMS-1} & \cdots & x_{i-1}^{HMS-1} & x_i^{HMS-1} \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_{i-1}^{HMS} & x_i^{HMS} \end{bmatrix} \begin{matrix} \Rightarrow \\ \Rightarrow \\ \Rightarrow \\ \Rightarrow \end{matrix} \begin{bmatrix} f(x^1) \\ f(x^2) \\ \vdots \\ f(x^{HMS-1}) \\ f(x^{HMS}) \end{bmatrix} \quad (1)$$

Where:

$x_1, x_2, \ldots, x_{i-1}, x_i$	=	A candidate solution
i	=	The number of candidate solutions in each solution vector
$f(x^1), f(x^2), \ldots, f(x^{HMS})$	=	The corresponding penalty value (i.e., the fitness) of each solution vector

The HM is the equivalent of the genetic pool in GA (Geem *et al*., 2002).

Improvise new solution: In step 3, a new solution vector is improvised stochastically using the following three operators:

- Random selection
- Memory consideration
- Pitch adjustment Eq. 2 shows the selection mechanism of HSA:

$$x_i^{New} \leftarrow \begin{cases} x_i(k) \in \{x_i(1), x_i(2), \ldots, x_i(k_i)\} & \text{w.p. } P_{Random} \\ x_i(k) \in \{x_i^1, x_i^2, \ldots, x_i^{HMS}\} & \text{w.p. } P_{Memory} \\ x_i(k \pm m) & \text{w.p. } P_{Pitch} \end{cases} \quad (2)$$

The value of variable i (i = 1, 2, ..., n) can be randomly selected from the set of all candidate discrete values $\{x_i(1), x_i(2), \ldots, x_i(k_i)\}$ with a probability of P_{Random} = (random selection) or it can be selected from the set of good values stored in the HM with a probability of P_{Memory} = (memory consideration) with slightly adjustment by moving to neighboring values of x_i (k±m) with a probability of P_{Pitch} = (pitch adjustment).

Update the harmony memory: In step 4, the new solution x_i^{new} is evaluated based on the total penalty value associated due to violating of the soft constraints during the improvisation step. Therefore, based on that penalty if it is lower than the total penalty of the worst solution, HMS-1 then the HMS-1 solution is removed from the HM and replaced by the new candidate solution x_i^{new}. Then, the HM is sorted in order to allocate the New Harmony (solution) in the right position. Otherwise x_i^{new} is discarded.

Check for stopping criteria: In step 5, researchers repeat step 3 and 4 until reaching the stopping criterion. Generally, the stopping criterion of HSA is when reaching the maximum Number of Improvisations (NI).

Outputting the best solution (Cadenza): In musical process, the musical passage that occurred at the end of a movement called Cadenza. In the context of the HSA, cadenza represents the process taking place at the end of the HSA computing. In this process, HSA returns the best quality solution ever found.

Modification on HSA parameters: Since, basic HSA requires proper parameter setting to ensure the effectiveness of HSA search many researchers focused on modifying the HSA parameter. These modifications involve parameter initialization, improvisation step and stopping criterion stopping criterion. Slow convergence may occur due to improper parameter setting. Those settings need to be carefully assigned depending on the problem/instances to be solved.

Modifications to HM initialization: In the basic HSA, the second step after initializing the parameters is to randomly generate a population of solution vectors to fill up the HM. In the literature of HSA some attempts were reported to improve the step of initializing the HM. Degertekin (2008) modified HM initialization step by generating solution vectors twice of the HMS then filling up the HM with the best solution vectors. Wang and Huang (2010) made another attempt by using low-discrepancy sequences instead of random mechanism to initialize the HM.

Modifications to improvisation step: Some modifications to the improvisation step have appeared in the literature of HSA. Geem *et al.* (2005) introduced the idea of multiple PAR strategy to solve generalized orienteering problem. They proposed three PARs, namely:

- The rates of moving to nearest
- Second nearest
- Third nearest cities

Among the earlier studies that attempted to solve the issue of using fixed parameter values of HSA is performed by Mahdavi *et al.* (2007) who proposed to use dynamic values of parameters settings of PAR and BW instead of using fixed PAR and BW of basic HSA. The proposed algorithm is called Improved Harmony Search (IHS). Equation 3 and 4 show the dynamic updating of PAR and BW equations, respectively (Mahdavi *et al*., 2007):

$$PAR(g) = PAR_{min} + \frac{PAR_{max} - PAR_{min}}{NI} \times g \qquad (3)$$

$$BW(g) = BW_{max} \exp\left(\frac{\ln\left(BW_{min}/BW_{max}\right)}{NI} \times g\right) \qquad (4)$$

Whereas:

PAR (g) and BW (g) = The pitch adjustment rate and the distance bandwidth in generation g, respectively NI is the maximum number of iterations g is the current iteration

PAR_{min} and PAR_{max} = The minimum and the maximum pitch adjustment rate, respectively

BW_{min} and Bw_{max} = The minimum and the maximum bandwidth, respectively

Another idea to improve IHS is proposed by Omran and Mahdavi (2008). They have developed another variant of HSA that was inspired by the idea of swarm intelligence. The new algorithm is called the Global best Harmony Search (GHS). GHS differs from IHS in the step of improvisation where they modified the pitch adjustment rule. To improvise new solution using GHS a new harmony will only be affected by the best harmony in the harmony memory. BW is removed from GHS. Equation 5 shows the rule of adjust pitches in GHS:

$$X_{new}(j) = X_B(k),\ j = 1, 2, ..., n \text{ and } k = Rand(1;n) \qquad (5)$$

Whereas:

X_{new} = The new harmony

X_B = The best harmony in harmony memory

k = A random integer between 1 and n

Dos Coelho *et al*. (2009) introduced another modified version of HSA based on the idea IHS by Mahdavi *et al*. (2007). They included the grade of the solution vectors into the Eq. 3 of PAR in IHS. Equation 6 shows the grade updating equation:

$$Grade = \frac{F_{max}(g) - mean(F)}{F_{max}(g) - F_{min}(g)} \qquad (6)$$

In this equation, F_{max} (g) and F_{min} (g) are the maximum and minimum of objective function values in generation g, respectively. Mean (F) is the means of the objective function value of the HM. Dos Coelho *et al*. (2009) proposed a new equation of PAR as shown in Eq. 7:

$$PAR(g) = PAR_{min} + \frac{PAR_{max} - PAR_{min}}{NI} \times g \times grade \qquad (7)$$

Another essential modification to the improvisation step were introduced by Saka and Hasancebi (2009) and Hasancebi *et al*. (2010) to change the parameter values of HMCR and PAR instead of changing the values of PAR and BW. Equation 8 and 9 show the dynamic calculating of these parameters:

$$HMCR^k = \left(1 + \frac{1-(HMCR)'}{(HMCR)'}.e^{-y.N(0,1)}\right)^{-1} \qquad (8)$$

$$PAR^k = \left(1 + \frac{1-(PAR)'}{(PAR)'}.e^{-y.N(0,1)}\right)^{-1} \qquad (9)$$

where, $HMCR^k$ and PAR^k are the sampled values of the adapted parameters for a new solution vector. N (0, 1) is a distributed random value γ is the learning rate of adapted parameters which is recommended to be selected within a range of ∈ (0.25 and 0.50). In the initialization step of HM the value of zero is set to the $HMCR^{(0)}$ and $PAR^{(0)}$ then these values will be updated according to Eq. 8 and 9, respectively.

Al-Betar and Khader (2010) proposed a Multi-PAR strategy to improve the performance of HSA in solving course time tabling problem. Eight local search procedures were applied to change the new harmony rather than using one PAR value.

Modification to stopping criterion of HSA: In metaheuristic approaches, there are many stopping criteria such as:

- Terminating the computation after a given number of iterations
- Reaching maximum computational time
- Reaching predefine value of the objective function
- when there is no improvement in the objective function value for a specified number of iterations

The standard stopping criterion in basic HSA is when reaching maximum Number of Improvisation (NI) that is defined during initiation stage.

A little research can be found in the literature with regard to proposing new stopping criterion of HSA. Kattan *et al*. (2010) used IHS for feed-forward Artificial

Table 1: Hybridization of HSA with components from other metaheuristics

Type of hybridization	Description	References
HSA+SA	He modified the PAR parameter using the cooling strategy of SA	Taherinejad (2009)
HSA+PSO	The PSO concept and global best particle are incorporated by replacing the BW parameter altogether and adding a randomly selected decision variables from the best harmony vector in HM	Omran and Mahdavi (2008)
HSA+PSO	The PSO concept and global best particle are used to improve the selection process in Harmony Memory Consideration Operator (HMCR)	Geem (2009)
HSA+DPSO	Dynamic PSO component is introduced to dynamically update the value of PAR parameter	Dos Coelho *et al.* (2009)
HSA+GA	Roulette-Wheel memory consideration is used to improve the selection process in HMCR	Al-Betar and Khader (2010)
HSA+CSA	The CSA (Clonal Selection Algorithm) is used to fine tune all HM vectors and improve the convergence capability of HS	Wang *et al.* (2009)
HSA+GA+SA+AIS	The hybridization aimed to enhance the solutions stored in HM to speed up the convergence and to prevent the HSA from getting stuck in the local optimal problem	Lee and Zomaya (2009)
HSA+PSO+GA	The hybridization aimed to make HSA as a global optimization algorithm by adding two operations: position updating and genetic mutation	Zou *et al.* (2011)
HSA+FCM	FCM is integrated in HSA to improve its convergence speed and fine tune the clustering quality	Alia *et al.* (2010)
HSA+K-means	k-means is used as a local search component in HAS	Mahdavi and Abolhassani (2009)
IHS+FCM	FCM is integrated into IHS to improve its local search ability and fine tuning the clustering result as a final step	Malaki *et al.* (2008)
HSA+Solver	Solver is used to support the exploitation mechanism of HAS	Ayvaz (2009)
HSA+NM-SA	It is used to improve the local search ability of HSA	Jang *et al.* (2008)
HSA+Taguchi	It is used to improve the initialization step for harmony memory and to reduce the effects of noise factors	Yildiz and Ozturk (2010)
SA+DE	DE (Differential Evolution) is used to fine tune the HM vectors and for multi-modal problems they proposed a new harmony memory updating strategy	Gao *et al.* (2009)

Table 2: Hybridizing HSA components and concepts in other metaheuristics

Type of hybridization	Description	References
PSO+HSA	The (HM) concept in HSA is integrated into PSO algorithm to prevent the pbest concept of PSO to violate the variables' boundary	Li *et al.* (2007)
PSOPC+ACO+HAS	HM concept is used to control the variable constraints in PSOPC	Kaveh and Talatahari (2009)
GA+HSA	The concept of selecting the decision variables from all vectors stored in the HM is mimicked to improve the GA selection mechanism	Li *et al.* (2008)
GA+HSA	HSA is used to maintain a balance between the exploration and exploitation concepts in GA	Nadi *et al.* (2010)
LDA+HSA	HSA is used as a pre-processing technique to overcome the LDA's problem	Moeinzadeh *et al.* (2009)

Neural Networks (ANN) as a new training technique. They used a modified strategy based on Best to Worst (BtW) harmony ratio in the current HM instead of using the standard stopping criterion where HSA stops when reaching the maximum Number of Improvisations NI.

Another stopping criterion is presented by Fourie *et al.* (2010) where they measured the distance between the best and worst solution in the HM. They also used a trade-off between speed and accuracy to stop the current process and move to the next phase (Fourie *et al.*, 2010).

Hybridization of HSA with other metaheuristics: The hybridization between components of metaheuristic approaches to improve the performance of HSA is found in the literature of metaheuristic and recommended by many researchers such as Blum *et al.* (2011) and Talbi (2009). HSA has been hybridized with other meta heuristic components such as Particle Swarm Optimization (PSO) (Geem, 2009; Omran and Mahdavi, 2008) Genetic Algorithm (GA) (Al-Betar and Khader, 2010) Simulated Annealing (SA) (Sui *et al.*, 2010) and others. Table 1 shows some research on hybridization of HSA with metaheuristics. This table is adopted from the survey of HSA variants by Alia and Mandava (2011).

One the other hand, HSA components have been used in other metaheuristic approaches such as in Li and Li (2007). The mechanism of selecting the decision variables of HSA was used to improve the selecting mechanism of GA such as by Li *et al.* (2008). Table 2 shows some examples of metaheuristic approaches that used some concepts and components of HSA.

CONCLUSION

Harmony Search Algorithm (HSA) has proved its ability to solve difficult optimization problems such as university course timetabling vehicle routing and other optimization problems. However not much research was reported on applying HSA to solve NRP. The motivation of studying harmony search algorithm by many researchers is to highlight the strength of HSA that is based on combining some local search components in the population based algorithm. Different issues and aspects related to HSA have been discussed concerning new variant proposed to overcome the weakness of basic harmony search algorithm. Many variants of HSA have been proposed by hybridizing some concepts of other metaheuristic approaches to enhance the HSA to employing some components of HSA to improve the other metaheuristic approaches. This study had reviewed research issues of the parameter settings in HSA and application of HSA to effectively solve some complex problems. HSA combines different characteristics of existing metaheuristic algorithms by using the harmony memory (to preserve the history of previous explored solutions) managing several solution vectors simultaneously and the dynamic change of the parameters of HSA during the searching process.

ACKNOWLEDGEMENT

Researchers wish to thank Ministry of Higher Education for supporting this research under the FRGS Research Grant Scheme (FRGS/1/2012/SG05/UKM/02/11).

REFERENCES

Al-Betar, M.A. and A.T. Khader, 2010. A harmony search algorithm for university course timetabling. Ann. Oper. Res., 194: 3-33.

Alia, O.M. and R. Mandava, 2011. The variants of the harmony search algorithm: An overview. Artif. Intell. Rev., 36: 49-68.

Alia, O.M., R. Mandava and M.E. Aziz, 2010. A hybrid harmony search algorithm to mri brain segmentation. Proceedings of the 9th IEEE International Conference on, Cognitive Informatics (ICCI), 2010, July 7-9, 2010, Beijing, pp: 712-721.

Ayvaz, M.T., 2009. Application of harmony search algorithm to the solution of groundwater management models. Adv. Water Resour., 32: 916-924.

Blum, C., J. Puchinger, G.R. Raidl and A. Roli, 2011. Hybrid metaheuristics in combinatorial optimization: A survey. Applied Soft Comput., 11: 4135-4151.

Degertekin, S.O., 2008. Optimum design of steel frames using harmony search algorithm. Struct. Multidi. Optim., 36: 393-401.

Dos Coelho, L.S. and D.L. de Andrade Bernert, 2009. An improved harmony search algorithm for synchronization of discrete-time chaotic systems. Chaos, Solitons, Fractals, 41: 2526-2532.

Fourie, J., S. Mills and R. Green, 2010. Harmony filter: A robust visual tracking system using the improved harmony search algorithm. Image Vision Comput., 28: 1702-1716.

Gao, X.Z., X. Wang and S.J. Ovaska, 2009. Uni-modal and multi-modal optimization using modified harmony search methods. Inter. J. Innovative Comput. Inf. Control, 5: 2985-2996.

Geem, Z.W., 2006. Optimal cost design of water distribution networks using harmony search. Eng. Optim., 38: 259-280.

Geem, Z.W., 2009. Particle-swarm harmony search for water network design. Eng. Optim., 41: 297-311.

Geem, Z.W., 2010. Recent Advances in Harmony Search Algorithm. 1st Edn., Springer, Berlin Heidelberg.

Geem, Z.W., J.H. Kim and G.V. Loganathan, 2002. Harmony search optimization: Application to pipe network design. Int. J. Model. Simul., 22: 125-133.

Geem, Z.W., J.H. Kim and G.V. Loganathan, 2001. A new heuristic optimization algorithm: Harmony search. Simulation, 76: 60-68.

Geem, Z.W., K.S. Lee and Y. Park, 2005. Application of harmony search to vehicle routing. Am. J. Applied Sci., 2: 1552-1557.

Glover, F. and M. Laguna, 1993. Tabu Search. In: Modern Heuristic Techniques for Combinatorial Problems. Reeves, C.R. (Ed.). Blackwell Scientific Publishing, Oxford, UK., pp: 70-150.

Hasancebi, O., F. Erdal and M.P. Saka, 2010. An adaptive harmony search method for structural optimization. J. Struct. Eng., 136: 419-431.

Holland, J.H., 1975. Adaptation in Natural and Artificial System. University of Michigan Press, Ann Arbor, USA.

Ingram, G. and T. Zhang, 2009. Overview of Applications and Developments in the Harmony Search Algorithm. In: Music-Inspired Harmony Search Algorithm, Geem, Z.W. (Ed.). Springer, Berlin/Heidelberg, pp: 15-37.

Jang, W.S., H. Kang and B.H. Lee, 2008. Hybrid simplex-harmony search method for optimization problems. Proceedings of the IEEE Congress on Evolutionary Computation, June 1-6, 2008, Hong Kong, pp: 4157-4164.

Kattan, A., R. Abdullah and R. Abdul Salam, 2010. Harmony search based supervised training of artificial neural networks. Proceedings of the International Conference on Intelligent Systems, Modelling and Simulation, January 27-29, 2010, Liverpool, pp: 105-110.

Kaveh, A. and S. Talatahari, 2009. Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. Comput. Struct., 87: 267-283.

Kirkpatrick, S., C.D. Gelatt Jr. and M.P. Vecchi, 1983. Optimization by simulated annealing. Sci., 220: 671-680.

Lee, K.S. and Z.W. Geem, 2004. A new structural optimization method based on the harmony search algorithm. Comput. Struct., 82: 781-798.

Lee, Y.C. and A.Y. Zomaya, 2009. Interweaving heterogeneous metaheuristics using harmony search. Porceedings of the IEEE International Symposium on Parallel and Distributed Processing, May 23-29, 2009, Rome, pp: 1-8.

Li, H.Q. and L. Li, 2007. A novel hybrid particle swarm optimization algorithm combined with harmony search for high dimensional optimization problems. Porceedings of the 2007 International Conference on Intelligent Pervasive Computing, October 11-13, 2007, Jeju City, pp: 94-97.

Li, M.J., M.K. Ng, Y.M. Cheung and J.Z. Huang, 2008. Agglomerative fuzzy K-means clustering algorithm with selection of number of clusters. Knowl. Data Eng., 20: 1519-1534.

Li, Q., N. Mitianoudis and T. Stathaki, 2007. Spatial kernel k-harmonic means clustering for multi-spectral image segmentation. Image Proc., 1: 156-167.

Mahdavi, M. and H. Abolhassani, 2009. Harmony k-means algorithm for document clustering. Data Min. Knowl. Discovery, 18: 370-391.

Mahdavi, M., M. Fesanghari and E. Damangir, 2007. An improved harmony search algorithm for solving optimization problems. Applied Math. Comput., 188: 1567-1579.

Moeinzadeh, H., E. Asgarian, M. Zanjani, A. Rezaee and M. Seidi, 2009. Combination of harmony search and linear discriminate analysis to improve classification. Proceedings of the 3rd Asia International Conference on Modelling and Simulation, May 25-29, 2009, Bali, pp: 131-135.

Nadi, F., A.T. Khader and M.A. Al-Betar, 2010. Adaptive genetic algorithm using harmony search. Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, ACGEC'10, pp: 819 820.

Omran, M.G. and M. Mahdavi, 2008. Global-best harmony search. Applied Math. Comput., 198: 643-656.

Sui, J., L. Yang, H. Fan and Z. Hua, 2010. Mine airflow optimizing control based on harmony annealing search. Proceedings of the International Conference of Information Science and Management Engineering, August 7-8, 2010, Xi'an, pp: 295-298.

Taherinejad, N., 2009. Highly reliable harmony search algorithm. Proceedings of the European Conference on Circuit Theory and Design, August 23-27, 2009, Antalya, pp: 818-822.

Talbi, E.G., 2009. Metaheuristics: From Design to Implementation. John Wiley and Sons, New York, USA., ISBN-13: 978-0470278581, Pages: 624.

Wang, C.M. and Y.F. Huang, 2010. Self-adaptive harmony search algorithm for optimization. Expert Syst. Appli., 37: 2826-2837.

Wang, X., X.Z. Gao and S.J. Ovaska, 2009. Fusion of clonal selection algorithm and harmony search method in optimisation of fuzzy classification systems. Int. J. Bio-Inspired Comput., 1: 80-88.

Yildiz, A. and F. Ozturk, 2010. Hybrid Taguchi-Harmony Search Approach for Shape Optimization. In: Recent Advances in Harmony Search Algorithm, Zw, G. (Ed.). Springer, Berlin, pp: 89-98.

Zou, D., L. Gao, S. Li and J. Wu, 2011. An effective global harmony search algorithm for reliability problems. Expert Syst. Appli., 38: 4642-4648.