Harmony Search Algorithm Based on Cloud Theory

Lifu Wang, Zhi Kong, Xingang Wang
College of Control Engineering, Northeastern University at Qinhuangdao, Qinhuangdao, China
e-mail: wlfkz@qq.com

***Abstract*—The parameter bw is a distance bandwidth between 0 and 1 in harmony search algorithm, which helps the algorithm in finding globally and locally improved solution. In this paper, a new harmony search algorithm is proposed in order to improve the convergence speed and the global search algorithm of the harmony search algorithm. The parameter bw is adjusted based on cloud theory in this new algorithm. The new algorithm is tested on some benchmark functions and the results are compared with the result of the traditional harmony search. Experimental results indicate that the new harmony search algorithm has a good performance in the global search ability and convergent speed.**

Keywords- cloud theory; harmony search; optimization; the bandwith (bw)

I. INTRODUCTION

Traditional optimization methods can be classified into two distinct group; direct and gradient-based methods. These two methods have their own limits. Recently, Geem et al. [1] developed a new harmony search (HS) algorithm which is inspired from musicians. The harmony search algorithm conceptualizes a behavioral phenomenon of musicians in the improvisation process, where each musician continues to experiment and improve this or her contribution in order to search for a better state of harmony[2][3]. However, harmony search algorithm is not efficient in performing local search for applications, Kong et al. introduced adaptive harmony search algorithm[4]. Mahdavi et al. proposed an improved harmony search algorithm (IHS) [5].

Clouds are a ubiquitous feature of our world[6]. Since the cloud model [7] is proposed, researchers put more attentions on it. Especially the cloud theory is applied in optimization to adjust the parameters[8-11].

In this paper, a new harmony search algorithm is proposed based on cloud theory. The traditional harmony search algorithm uses fixed value for bw, which may effect the convergence degree of solution. The parameter bw in the new harmony search algorithm based on cloud theory is adjusted dynamically to improve the convergence speed and the global search ability. And the new algorithm is tested on some benchmark functions. Experimental results indicate that the new harmony search algorithm has a good performance in the global search ability and convergent speed.

II. HARMONY SEARCH ALGORITHM

Current meta-heuristic algorithms imitate natural phenomena, i.e., physical annealing in simulated annealing, human memory in tabu search, and evolution in evolutionary algorithms. A harmony search algorithm was conceptualized using the musical process of searching for a perfect state of harmony. Musical performances seek to find pleasing harmony (a perfect state) as determined by an aesthetic standard, just as the optimization process seeks to find a global solution (a perfect state) as determined by an objective function. The pitch of each musical instrument determines the aesthetic quality, just as the objective function value is determined by the set of values assigned to each decision variable. Figure 1 shows the details of the analogy between music improvisation and engineering optimization. In music improvisation, each player sounds any pitch within the possible range, together making one harmony vector. If all the pitches make a good harmony, that experience is stored in each player's memory, and the possibility to make a good harmony is increased next time. Similarly in engineering optimization, each decision variable initially chooses any value within the possible range, together making one solution vector. If all the values of decision variables make a good solution, that experience is stored in each variable's memory, and the possibility to make a good solution is also increased next time[3]. The optimization procedure of the HS algorithm consists of Steps 1 through 5[1]:

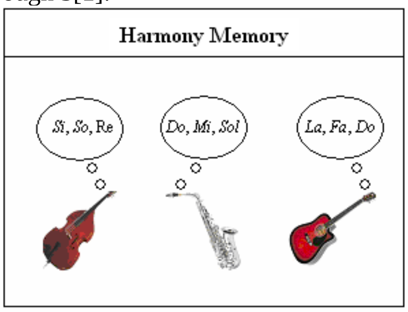

Figure 1. Structure of harmony memory

Step 1. Initialize the optimization problem and algorithm parameters.

Step 2. Initialize the harmony memory (HM).

Step 3. Improvise a new harmony from the HM.

Step 4. Update the HM.

Step 5. Repeat Steps 3 and 4 until the termination criterion is satisfied.

1. Initialize the Optimization Problem and Algorithm Parameters

In this step, the optimization problem is specified as follows:

Minimize $z = f(x)$ subject to $x_i \in X_i \quad i = 1, 2, \cdots, N$ (1)

where $f(x)$ is the objective function to be minimized, x_i are the decision variables, X_i are the set of possible ranges of each variable, and N is the number of decision variables. Initialize the value HMS (harmony memory seize), HMCR (harmony memory considering rate), PAR(pitch adjusting rate), bw (bandwidth) and the termination criterion.

2. Initialize the Harmony Memory (HM)

In this step, HM matrix is filled with randomly generated solution vectors as many as the HMS and corresponding fitness function values are calculated as

$$\begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_{N-1}^1 & x_N^1 \\ x_1^2 & x_2^2 & \cdots & x_{N-1}^2 & x_N^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{HMS-1} & x_2^{HMS-1} & \cdots & x_{N-1}^{HMS-1} & x_N^{HMS-1} \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_{N-1}^{HMS} & x_N^{HMS} \end{bmatrix} \Rightarrow \begin{bmatrix} f(x^1) \\ f(x^2) \\ \vdots \\ f(x^{HMS-1}) \\ f(x^{HMS}) \end{bmatrix} \quad (2)$$

3. Improvise a New Harmony From the HM

After the memory consideration, each decision variable is evaluated to determine whether pitch adjustment is necessary or not. This evaluation is carried out with PAR parameter which is the probability of pitch adjusting and identified as follows:

$$x_i' = \begin{cases} x_i' \pm \text{Rnd}(0;1) \times \text{bw with probability } PAR \\ x_i' \text{ with probability } 1\text{-}PAR \end{cases} \quad (3)$$

In Eq. (3), bw is an arbitrary bandwidth, Rnd(0;1) is a uniform random number between 0 and 1.

4. Update the Harmony Memory

In this step, the comparison between new harmony vector $x' = (x_1', x_2', \cdots, x_N')$ and worst harmony in the HM is performed in terms of their objective function values. If the new harmony vector is better than worst harmony, the new harmony vector is included to the HM and worst harmony is excluded.

5. Check the Termination Criterion

In this step, the optimization process continues with the computation by iterating Steps 3-5 until the given termination criterion is satisfied.

Figure 2 shows the optimization procedure of the harmony search algorithm.

III. THE IMPROVED HARMONY SEARCH ALGORITHM

A. Normal Cloud Model[12][13]

The cloud model is a model of the uncertain transition between a linguistic term of a qualitative concept and its numerical representation. The cloud model is as follows:

Let U be the set $U = \{u\}$, as the universe of discourse, and T a linguistic term associated with U. The membership degree of u in U to the linguistic term T, $C_T(u)$, is a random number with a stable tendency. $C_T(u)$ takes the values in [0,1]. A compatibility cloud is a mapping from the universe of discourse U to the unit interval [0,1]. That is, $C_T(u): U \to [0,1]$, $\forall u \in U$, $u \to C_T(u)$. Namely, the random distribution of $C_T(u)$ in U is called (membership) cloud. When $C_T(u)$ obeys the normal distribution, cloud is called normal cloud model.

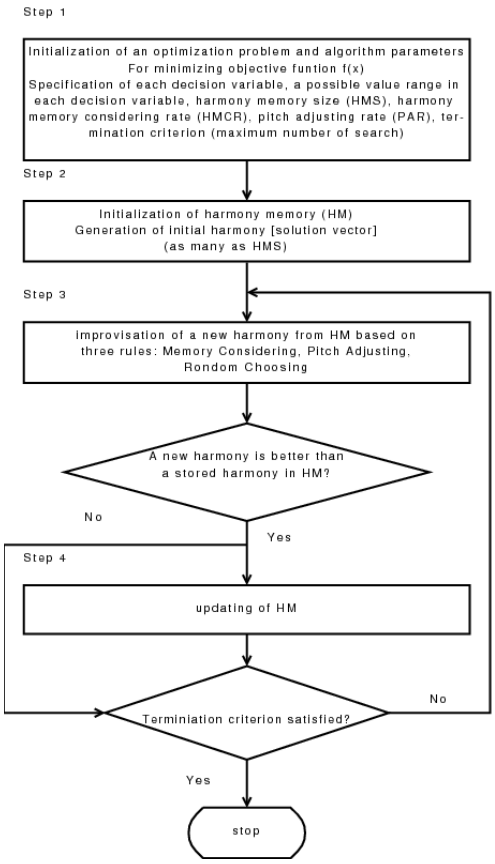

Figure 2. Optimization procedure of the harmony search algorithm

A normal cloud is defined with three digital characteristics, expected value E_x, entropy E_n, and hyper-entropy H_e. The expected value E_x is the position at U corresponding to the center of gravity of the cloud. The entropy E_n is a measure of the coverage of the concept within the universe of discourse. The hyper-entropy H_e is the entropy of the entropy of the entropy E_n. It is a measure of dispersion of the cloud drops.

B. Harmony search algorithm based on cloud theory

The bandwidth (bw) is an important parameter in harmony search algorithm. The traditional harmony search algorithm uses fixed value for bw. However, bw should be decreased when all objective value are centralized and increased when all objective values are scattered in the solution space. In this paper, we introduce cloud theory to adjust the value bw in the harmony search to improve the global search ability and make faster convergence speed of the algorithm.

The optimization procedure of the improved harmony search algorithm consists of 5 steps, which is similar to the traditional harmony search algorithm. However, the value of bw is different from that of the traditional harmony search algorithm.

For the value bw, it can be obtained as follows:

1. Set the maximum of bw is $bw_{\max}$, the minimum of bw is $bw_{\min}$.

1. Calculate the fitness function value $f(x^i)$ for each solution vector.

2. Calculate the average fitness F_{ave} and the best fitness F_{best} using the following equations:

$$F_{ave} = \sum_{i=1}^{HMS} f(x^i) / HMS$$

$$F_{best} = \min\{f(x^i)\}$$

3. Select a variable x_i' in the harmony memory corresponding to the fitness value $f(x_i')$.

if $f(x_i') < F_{best}$

bw= $bw_{\min}$

elseif $f(x_i') > F_{best}$

bw= $bw_{\max}$

else

$$E_x = F_{best}$$

$$E_n = (f(x_i') - F_{best}) / c_1$$

$$H_e = E_n / c_2$$

$$E_n' = normrnd(E_n, H_e)$$

$$bw = bw_{\max} - bw_{\min} \exp\{-\frac{(f(x_i') - F_{best})}{2E_n'^2}\}$$

end

where c_1 and c_2 are parameters. Generally, $c_1 = 3$, $c_2 = 10$.

IV. SIMULATION RESULTS

This section compares the performance of the harmony search algorithm based on cloud theory (HSC) with that of the adaptive harmony search algorithm (AHS), improved harmony search algorithm (IHS), and traditional harmony search algorithm (HS). For the HSC, HMS=10, HMCR=0.85, PAR=0.8, bw_{max}=0.02, bw_{min}=0.001. For HM, HMS=10, HMCR=0.85, PAR=0.8, bw=0.05. For IHS, HMS=10, HMCR=0.85, PAR_{max}=0.95, PAR_{min}=0.35, bw_{max}=0.02, bw_{min}=0.001. For AHS, HMS=10, HMCR=0.85, c_1=1, c_2=0.5. All functions were implemented in 10 dimensions except for the two-dimensional Schaffer function.

The harmony size is set to 10 in harmony search algorithms. Five prevalent benchmarks described in [14] are employed, and their configuration is listed in Table 1. The experimental results are listed through Figures 3-7.

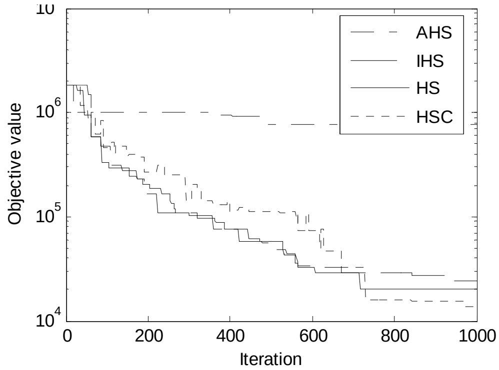

Figure 3. Sphere function

TABLE I. Benchmark configuration for simulation

Function	Name	Domain	Minimum
f1	Sphere	[-1000,1000]	0
f2	Rosenbrock	[-30,30]	0
f3	Schaffer	[-5.12,5.12]	0
f4	Ackley	[-30,30]	0
f5	Griewank	[-600,600]	0

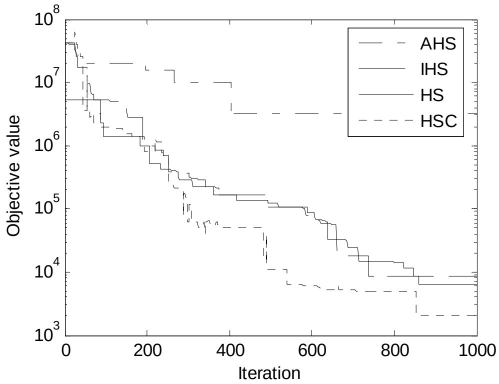

Figure 4. Rosenbrock function

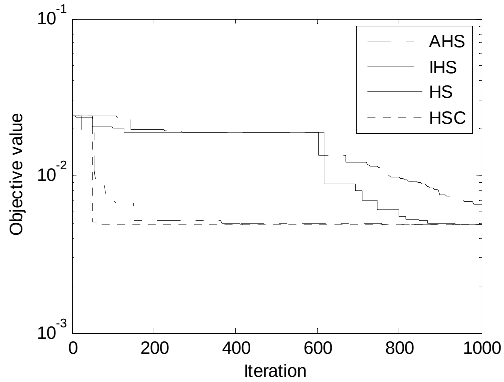

Figure 5. Schaffer function

The benchmark functions as follows:

Sphere:

$$f_1 = \sum_{i=1}^{n} x_i^2$$

Rosenbrock:

$$f_2 = \sum_{i=1}^{n} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$$

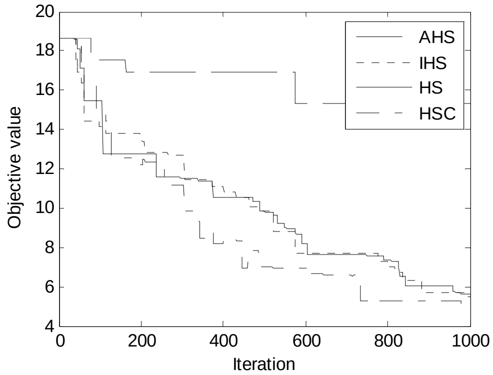

Figure 6. Ackley function

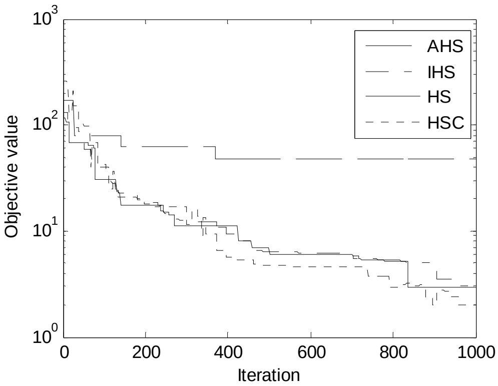

Figure 7. Griewank function

Schaffer:

$$f_3 = 0.5 + \frac{\sin^2(\sqrt{x^2 + y^2}) - 0.5}{(1 + 0.001(x^2 + y^2))^2}$$

Ackley:

$$f_4 = -20\exp\left(-0.2\sqrt{1/n\sum_{i=1}^{n} x_i^2}\right) - \exp\left(1/n\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$$

Griewank:

$$f_5 = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

In order to compare four algorithms distinctively, log-scale vertical and horizontal ordinates are used to denote the objective function value and iteration times, in Figure 3-5 and Figure 7. From Figure 5, we can see HSC algorithm rapidly converges to the global optimum. From Figure 3, Figure 4, Figure 6, and Figure 7 HSC algorithm has the better precision error than other algorithms.

V. CONCLUSIONS

This paper has introduced a harmony search algorithm based on cloud theory to adjust parameter bw in harmony search algorithm, whereas the HS algorithm on consider the fixed value bw. The dynamic parameter bw in this paper can increase the flexibility of the HS algorithm and produce better solutions. And several benchmarks are used to test HSC algorithm. The results are better than those previously reported in the literature.

ACKNOWLEDGMENT

This work is supported by the Natural Science Foundation of Hebei under Grant No. F2012501030, the Fundamental Research Funds for the Central Universities under Grant No. N100323012, No. N100323011, and National Natural Science Foundation of China under Grant No. 51105068.

REFERENCES

[1] Z. W. Geem, J. H. Kim and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony search," Simulation, vol.76, no.2, pp.60-68, 2001.

[2] S.L. Kang, Z. W. Geem. "A new structural optimization method based on the harmony search algorithm". Comput Struct, vol.82, no.89, pp.781-789, 2004.

[3] K. S. Lee, Z. Geem. "A new meta-heuristic algorithm for continues engineering optimization: harmony search theory and practice". Comput Methods Appl Mech Eng, vol.194, pp.3902-3933, 2005.

[4] Z. Kong, L.Q. Gao, L.F. Wang, Y.F. Ge, "On Adaptive Harnony Search Algorithm," International Journal of Innovative Computing, Information and Control, vol. 5, no.9, pp.2551-2560, 2009.

[5] M. Mahdavi, M. Fesanghary and E. Damangir, "An improved harmony search algorithm for solving optimization problem", Applied Mathematics and Computation, vol.188, pp.1567-1579, 2007

[6] Mark J. Harris, William V.Baxter III, Thorsten Scheuermann, Anselmo Lastra, "Thrsten Scheuermann.Simulation of Cloud Dynamics on Graphics Hardware". Graphics Hardware, 2003.

[7] D.Y. Li, "Uncertainty reasoning based on cloud models in controllers," Computer science and Mathematics with Application, vol. 35, no. 3, pp. 99-123, 1998.

[8] J. Zhoa, J. SUN, and W. Xu. "Quantum-behaved Particle Swarm Optimization with Normal Cloud Mutation Operator," Computational Intelligence and Software Engineering, 2009 Wuhan, pp. 1-4.

[9] G. Yan, and Z. Hao. "A Novel A tmosphere Clouds Model Optimi-zation Algorithm," International Conference on Computing, Measure-ment, Control and Sensor Network. 2012, Taiyuan, pp. 217-220.

[10] J. Wen and B. Cao. "A Modified Particle Swarm Optimizer Based on Cloud Model," Proceedings of the 2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics July, 2008, Xi'an, China, pp, 1238-1241.

[11] J. Wen, X. Wu, K. Jiang and B. Cao, "Particle Swarm Algorithm Based on Normal Cloud," IEEE World Congress on Computational Intelligence, Hangkong, China, pp.1492-1496.

[12] C. Dai, Y. Zhu, W. Chen, "Cloud Theory-Based Genetic Algorithm," Journal of Southwest Jiao Tong University, , vol. 41, no. 6, pp.729-732, 2006.

[13] D. Li., H. Meng, X. Shi. "Membership Clouds and Membership Cloud Generators". Journal of Computer Research and Development, vol. 32, no. 6, pp.15-20, 1995.

[14] S. Janson, M. Middendorf, "A hierarchical particle swarm optimizer and its adaptive variant", IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics, vol.35, no.6, pp.1272-1282, 2005.