Improved Harmony Search Algorithm for Global Optimization

Guojun Li, Hongyu Wang

Physics Department, Anshan Normal University, Anshan 114005, China
E-mail:wlsyslgj@163.com why@btitgroup.com

Abstract: Harmony search (HS) algorithm, applied in many fields, is a new population algorithm, which imitates magically the phenomenon of musical improvisation process. However, it has a potential shortage, which is easily trapped into local optima when searching for global optima. To solve this problem, a hybrid harmony search algorithm (HHS) is promoted, which is based on the conception of swarm intelligence. HHS employed a local search method to replace the pitch adjusting operation, and designed an elitist preservation strategy to modify the selection operation. Experiment results demonstrated that the proposed method performs much better than the HS and its improved algorithms (IHS, GHS and NGHS).

Key Words: Harmony search algorithm; swarm intelligence; local search; elitist preservation

1 INTRODUCTION

Over the past decades, traditional methods such as liner programming, dynamic programming, conjugate gradient method, quasi-Newton method, etc. are usually applied to solve real-world problems. However, with the development of modern technique, more and more complex optimization problem are confronted in engineer design. The traditional methods can not solve or never find a satisfied solution for these optimization problems. And heuristic optimization algorithms which include include genetic algorithm (GA), ant colony algorithm (ACA), particle swarm optimizing algorithm (PSO) and differential evolution algorithm (DE), et al, are proposed and applied in recent years. The heuristic optimization algorithms imitating natural phenomena or process, which have been successfully applied to many real-world optimization problems [1-4]. Compared to the traditional optimization algorithm, Heuristic optimization algorithm has many advantages such as wide applicability, parallelism and global search ability, and that is the real motivation of heuristic optimization algorithm research.

Harmony search algorithm is a new meta-heuristic algorithm, which is proposed by Geem et al. [5] in 2001. Harmony search algorithm is inspired by the phenomenon of musician attuning. The HS algorithm has many merits. One highlight of the HS algorithm is that it generates a new vector or solution by considering all of the existing vectors or solutions, whereas the genetic algorithm only considers the two parent vectors or solutions [6-7]. Moreover, experiments demonstrated that the HS algorithm was faster than GA algorithm in runtime. Therefore, the HS algorithm has attached much attention and has been used with various science and engineering problems [8-12]. However, harmony search algorithm also suffers from a serious problem as other meta-heuristics does. It easily gets into trouble in performing local search for numerical applications. Thus, some improved variants were presented to improve the optimization performance of HS. These improved algorithms include: improved harmony search (IHS) algorithm [7], self-adaptive global-best harmony search (SGHS) algorithm [13], Global-best harmony search (GHS) algorithm [14], and novel global harmony search (NGHS) algorithm [15].

The paper is organized as follows. In Section 2, harmony search algorithm and its improved algorithms would be reviewed. In Section 3, a hybrid harmony search algorithm is introduced in detail. In section 4, global optimization problems are tested and make a comparison between the HHS and the other algorithms. Finally, conclusions were given in Section 5.

2 HS AND ITS IMPROVED ALGORITHM

2.1 The HS algorithm

In this subsection, we will overview harmony search algorithms. The original harmony search algorithm works as follows:

Step 1: Initialize the optimization problem and harmony search algorithm parameters. The optimization problem is defined as Minimize $f(x)$ subject to $x_{iL} \le x_i \le x_{iU}$, $(i=1,2,\cdots,N)$. x_{iL} and x_{iU} are the lower and upper bounds for decision variables, respectively. The HS algorithm parameters are specified in this step. They are the harmony memory size (*HMS*); harmony memory considering rate (*HMCR*); pitch adjusting rate *(PAR)*; and the number of improvisations (*K*), or stopping criterion.

Step 2: Initialize the harmony memory. The initial harmony memory (HM) is generated from a uniform distribution in the ranges $[x_{jL}, x_{jU}], (j=1,2,\cdots,N)$, showed as following:

This work is supported by National Nature Science Foundation under Grant 11275007, 11775090.

$$\mathrm{HM}=\left[\begin{array}{cccc|c} x_1^1 & x_2^1 & \cdots & x_N^1 & f(\vec{x}^1) \\ x_1^2 & x_2^2 & \cdots & x_N^2 & f(\vec{x}^2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_1^{\mathrm{HMS}} & x_2^{\mathrm{HMS}} & \cdots & x_N^{\mathrm{HMS}} & f(\vec{x}^{HMS}) \end{array}\right] \tag{1}$$

Step 3: Improvise a new harmony from the HM. Generating a new harmony is called improvisation. The new harmony vector x^{new} is determined by three rules: memory consideration, pitch adjustment and random selection. The procedure works as follows:

for $j=1$ to N **do**

if rand()≤$HMCR$

$x_j^{new}=x_j^{r}, (r\in(1,2,\cdots,HMS))$ %memory consideration

if rand()≤PAR **then**

$x_j^{new}=x_j^{new}\pm bw\times rand$ %pitch adjustment

endif

else $x_j^{new}=x_{jL}+rand\times(x_{jU}-x_{jL})$ %random selection

endif

endfor

Step 4: Update harmony memory. If the fitness of the improvised harmony vector x^{new} is better than that of the worst harmony, replace the worst harmony in the HM with x^{new}.

Step 5: Check the stopping criterion. If the maximal iteration number (K) is satisfied, computation is terminated. Otherwise, Steps 3 and 4 are repeated.

2.2 The Improved HS Algorithm

The IHS algorithm, GHS algorithm and NGHS algorithm will be introduced as follow.

IHS algorithm was proposed by M Mahdavi in 2007 [7]. In IHS algorithm, PAR and bw change dynamically with generation number as expressed as follow:

$$PAR(k)=PAR_{\min}+\frac{(PAR_{\max}-PAR_{\min})}{K}\times k \tag{2}$$

$$\mathrm{bw}(k)=\mathrm{bw}_{\max}\exp(c\cdot k),\ c=\frac{\ln\left(\frac{\mathrm{bw}_{\min}}{\mathrm{bw}_{\max}}\right)}{K} \tag{3}$$

Where K is the maximum number of iterations, and k is the current number of iterations; $\mathrm{PAR}_{\min}$ and $\mathrm{PAR}_{\max}$ are the minimum adjusting rate and the maximum adjusting rate, respectively; $\mathrm{bw}_{\min}$ and $\mathrm{bw}_{\max}$ are the minimum bandwidth and the maximum bandwidth, respectively. A large number of experiments and studies show that the IHS based on improved PAR and bw has better optimization performance than the HS in most cases.

The GHS modified the pitch-adjustment step of the HS such that the new harmony can mimic the best harmony in the HM [14], and it has exactly the same steps as the HS with the exception that Step 3 and the parameter PAR adjustment are modified as the same with IHS algorithm. In the GHS algorithm, the pitch adjusting procedure works as follows:

if rand()≤PAR **then**

$x_j^{new}=x_j^{\mathrm{best}}$ %where best is the index of the best harmony in the HM. pitch adjustment

endif

Inspired by the swarm intelligence of particle swarm, a new variation of HS, called NGHS, is proposed by Zou et al in 2010 year [15]. In NGHS algorithm, a new candidate harmony (represent solution) is generated by using position updating and genetic mutation with a low probability.

3 HHS ALGORITHM

3.1 Local Search Method

In HS algorithm, the pitch adjusting operation plays an important role in search process. However, to set a suitable value of bw is very difficult, so we proposed a local search method to replace the pitch adjusting operation. The local search works as follow:

Step 1 select m harmony vectors randomly and find the current best harmony vector x^{best} in the harmony memory.

Step 2 to calculate the mean value of these randomly selected harmony vectors, the computation equation is shown as follow:

$$\overline{x}_j=\sum_{i=1}^{m}x^i{}_j \tag{4}$$

Where x^i_j is the means the jth dimension of the ith harmony vector.

Step 3 employ the current best harmony vector x^{best} and the mean harmony vector $\overline{x}$ to search a new harmony vector x^{new}, which works as follow:

Inspir $x_j^{new}=x_j^{new}+rand_1\times(x_j^{\mathrm{best}}-round(1+rand_2)\times\overline{x}_j)$

Where rand1 and rand2 are the random number among [0,1], respectively; round represents get the near integer.

3.2 Elitist Preservation Strategy

To keep more useful information, an elitist preservation strategy is proposed in this subsection. We save the best harmony vector in a unique elitist harmony memory, named EHM. The size of the elitist harmony memory is fixed as the double of harmony memory size, that is EHM=2HM. In the initial step, the EHM also is initialized as the same with HM, and only the top best harmony vectors is preserved in EHM. In the iteration, the best harmony vector replaces the worst harmony vector of the EHM.

4 EXPERIMENTAL RESULTS AND ANALYSIS

To extensively investigate the performance of the HHS algorithm, 8 test functions are considered in the experiment and they are presented as follows:

Table1. The Parameter of Algorithms

Algorithm	HMS	HMCR	PAR/p_m	PAR_{min}
HS	5	0.95	0.3/—	—
IHS	5	0.95	—	0.01
GHS	5	0.95	—	0.01
NGHS	5	—	—/0.005	—
HHS	5	0.95	—	—

Table2. The Parameter of Algorithms

Algorithm	PAR_{max}	bw	bw_{min}	bw_{max}
HS	—	0.01	—	—
IHS	0.99	—	0.0001	$(x_{ju}-x_{jl})/20$
GHS	0.99	—	—	—
NGHS	—	—	—	—
HHS	—	—	—	—

Table3. Results of HSA for f_1-f_{10} (N=50)

Function	Algorithm	K	Best	Worst	Mean	SD
	HS	50000	6.8930e-001	2.3242e+001	6.1191e+000	4.6203e+000
	IHS	50000	1.4629e+00	2.2846e+004	1.8737e+004	1.9697e+003
f_1	GHS	30000	2.6100e-002	8.8100e-002	5.2400e-002	1.5400e-002
	NGHS	30000	3.1025e-003	2.8935e-002	1.0506e-002	2.0548e-002
	HHS	30000	**1.1615e-082**	**2.6736e-079**	**2.4658e-080**	**5.3287e-080**
	HS	50000	1.3143e+00	1.6463e+004	5.2425e+003	3.8963e+003
	IHS	50000	1.1101e+00	2.4732e+009	1.7362e+009	2.9635e+008
f_2	GHS	50000	2.3959e+00	1.0173e+004	1.4510e+003	2.5171e+003
	NGHS	50000	1.2463e+00	2.7684e+003	2.0859e+002	3.6831e+002
	HHS	30000	**9.2386e-002**	**3.1706e+000**	**1.0201e+000**	**8.6476e-001**
	HS	50000	1.8867e+00	2.9368e+001	2.4763e+001	2.4253e+000
	IHS	50000	5.7702e+00	7.1819e+002	6.4913e+002	4.3458e+001
f_3	GHS	30000	1.3700e+00	2.8751e+001	2.0660e+001	4,0128e+000
	NGHS	50000	8.2473e+00	2.5069e+001	1.5905e+001	4.1509e+000
	HHS	30000	**0**	**0**	**0**	**0**
	HS	50000	1.2452e+00	1.7491e+000	1.4037e+000	1.0220e-001
	IHS	50000	1.2555e+00	2.3245e+002	1.7590e+002	2.1940e+001
f_4	GHS	50000	2.0200e-002	1.0120e-001	4.7200e-002	2.4500e-002
	NGHS	50000	8.9000e-003	1.3280e-001	2.4800e-002	2.4200e-002
	HHS	30000	**0**	**0**	**0**	**0**
	HS	50000	1.2317e+00	1.2069e+000	1.6306e+000	2.1340e-001
	IHS	50000	1.2645e+00	1.3930e+001	1.3353e+001	3.2230e-001
f_5	GHS	50000	2.9400e-002	5.9200e-002	3.9700e-002	7.6400e-003
	NGHS	50000	1.9400e-003	3.9700e-002	2.6300e-002	4.5340e-003
	HHS	10000	**3.5527e-015**	**7.1054e-015**	**4.8554e-015**	**1.7413e-015**
	HS	50000	2.6320e-001	6.4420e-001	4.0690e-001	9.9500e-002
	IHS	50000	4.8923e+00	7.1292e+003	8.4979e+002	1.1900e+003
f_6	GHS	50000	9.2830e-001	1.9242e+000	1.2999e+000	2.4480e-001
	NGHS	50000	2.8758e-009	1.9179e-006	2.5790e-007	4.3448e-007
	HHS	50000	**0**	**0**	**0**	**0**
	HS	50000	1.2815e+00	2.0255e+003	1.6804e+003	2.0703e+002
	IHS	50000	1.2562e+00	1.9995e+003	1.5911e+003	1.5512e+002
f_7	GHS	50000	1.2443e+00	2.3718e+005	1.7077e+005	3.1502e+004
	NGHS	50000	1.1581e+00	2.0106e+005	1.5406e+005	2.2423e+004
	HHS	40000	1.4016e-007	7.4250e-006	2.7935e-006	2.1050e-006
	HS	50000	1.4402e+00	3.0654e+008	2.1856e+008	3.6949e+007
	IHS	50000	3.4322e+00	6.3573e+008	4.4958e+008	7.8263e+007
f_8	GHS	50000	3.4664e+00	6.4569e+007	4.6674e+007	6.7331e+006
	NGHS	50000	2.3974e+00	5.2648e+007	3.7092e+007	7.1979e+006
	HHS	40000	9.4233e-033	2.1815e-032	1.0353e-032	2.4609e-033

f_1 Sphere function, defined as $\min f_1 = \sum_{i=1}^{N} x_i^2$

Where global optimum $x^* = (0,0,...,0)$ and $f(x^*) = 0$ for $-100 \le x_i \le 100\ (i = 1,2,...,N)$.

f_2 Rosenbrock function, defined as

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

Where global optimum $x^* = (1,1,...,1)$ and $f(x^*) = 0$ for $-30 \le x_i \le 30\ (i = 1,2,...,N)$.

f_3 Rastrigrin function, defined as $\min f_3 = \sum_{i=1}^{N} (x_i^2 - 10\cos(2\pi x_i) + 10)$

Where global optimum $x^* = (0,0,...,0)$ and $f(x^*) = 0$ for $-10 \le x_i \le 10\ (i = 1,2,...,N)$.

f_4 Griewank function, defined as $\min f_4 = \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N} \cos(\frac{x_i}{\sqrt{i}}) + 1$

Where global optimum $x^* = (0,0,...,0)$ and $f(x^*) = 0$ for $-600 \le x_i \le 600\ (i = 1,2,...,N)$.

f_5 Ackley's function, defined as $\min f_5 = 20 + e - 20\exp(-0.2\sqrt{\frac{\sum_{i=1}^{N} x_i^2}{N}}) - \exp(\frac{\sum_{i=1}^{N}\cos(2\pi x_i)}{N})$

Where global optimum $x^* = (0,0,...,0)$ and $f(x^*) = 0$ for $-32 \le x_i \le 32\ (i = 1,2,...,N)$.

f_6 Schwefel's problem 2.22, defined as $\min f_6 = \sum_{i=1}^{N} |x_i| + \prod_{i=1}^{N} |x_i|$

Where global optimum $x^* = (0,0,...,0)$ and $f(x^*) = 0$ for $-100 \le x_i \le 100\ (i = 1,2,...,N)$.

f_7 Zakharov function, defined as

$$\min f_7 = \sum_{i=1}^{N} x_i^2 + \left(\sum_{i=1}^{N} 0.5ix_i^2\right)^2 + \left(\sum_{i=1}^{N} 0.5ix_i^2\right)^4$$

Where global optimum $x^* = (0,0,...,0)$ and $f(x^*) = 0$ for $-100 \le x_i \le 100\ (i = 1,2,...,N)$.

f_8 Rotated hyper-ellipsoid function, defined as $\min f_8 = \sum_{i=1}^{N} (\sum_{j=1}^{i} x_j)^2$

Where global optimum $x^* = (0,0,...,0)$ and $f(x^*) = 0$ for $-100 \le x_i \le 100\ (i = 1,2,...,N)$.

In the experiments, the parameter settings of compared HS algorithms are shown in Table 1 and Table 2. The best, worst, mean and standard deviation (SD) are reported over 30 independent simulations. For each simulation the procedure to be run in computer Inter(R) Pentium(R) 4, CPU 2.93GHz, and the numerical results using 5 harmony search algorithms are recorded in Table 3.

5 CONCLUSIONS

This paper presents a hybrid harmony search algorithm to improve the performance of HS algorithm. The HHS has two different aspects with the original HS algorithm: local search operation and elitist preservation strategy. Global optimization problems are carried out to be tested, the results obtained by the HHS algorithm are better than those obtained by HS, IHS, GHS and NGHS algorithm. The results also demonstrate the effectiveness and robustness of the proposed algorithm. In short, the HHS algorithm is a promising optimization algorithm, and it has high exploration capability of solution space throughout the whole iteration.

REFERENCES

[1] He Y H, Hui C W. A binary coding genetic algorithm for multi-purpose process scheduling: A case study. Chemical Engineering Science, 2010, 65(16): 4816-4828.

[2] Shi H X. Solution to 0/1 knapsack problem based on improved ant colony algorithm, in: International Conference on Information Acquisition, 2006, 1062-1066.

[3] Coelho L S. An efficient particle swarm approach for mixed-integer programming in reliability–redundancy optimization applications, Reliability Engineering & System Safety, 2009, 94(4): 830-837.

[4] Onwubolu G, Davendra D. Scheduling flow shops using differential evolution algorithm, European Journal of Operational Research, 2006, 171(2): 674-692.

[5] Geem Z W, Kim J H, Loganathan G V. A new heuristic optimization algorithm: harmony search. Simulation, 2001, 76(2): 60-68.

[6] Saka M P, Optimum design of steel sway frames to BS5950 using harmony search algorithm. Constructional Steel Research, 2009, 65(1): 36-43.

[7] M Mahdavi, M Fesanghary, E Damangir. An improve harmony search algorithm for solving optimization problems. Applied Mathematics and Computation 2007; 188(2) 1567-1579.

[8] Ayvaz M T. Application of Harmony Search algorithm to the solution of groundwater management models. Advances in Water Resources, 2009, 32(6): 916-924.

[9] Geem Z W, Lee K S, Park Y J. Application of harmony search to vehicle routing. American Journal of Applied Sciences, 2005, 12(2): 1552–1557.

[10] J H Kim, Z W Geem, E S Kim, Parameter estimation of the nonlinear Muskingum model using harmony search, Journal of American Water Resource Association, 2001, 37(5) 1131–1138.

[11] Geem Z W. Particle-swarm harmony search for water network design, Engineering Optimization, 2009, 41(4): 297-311.

[12] Sharma K D, Chatterjee A, Rakshit A. Design of a Hybrid Stable Adaptive Fuzzy Controller Employing Lyapunov Theory and Harmony Search Algorithm, IEEE Transactions on Control Systems Technology, 2010, PP(99): 1-8.

[13] Chia-Ming Wang, Yin-Fu Huang. Self-adaptive harmony search algorithm for optimization. Expert Systems with Applications, 2010, 37(4): 2826-2837.

[14] Omran M G H, Mahdavi M. Global-best harmony search, Applied Mathematics and Computation, 2008, 198(2): 643-656.

[15] Zou D X, Gao L Q, Wu J H, Li S. Novel global harmony search algorithm for unconstrained problems, Neuro-computing, 2010, 73(16-18): 3308-3318.