Xiao-Zhi Gao, Xiaolei Wang, and Seppo Jari Ovaska. Uni-modal and multi-modal optimization using modified harmony search methods. International Journal of Innovative Computing, Information and Control, in press.

International Journal of Innovative Computing, Information and Control
Volume **x**. Number **x. x** **2009**
ICIC International @2009 ISSN 1349-4198

UNI-MODAL AND MULTI-MODAL OPTIMIZATION USING MODIFIED HARMONY SEARCH METHODS

XIAO-ZHI GAO, XIAOLEI WANG AND SEPPO JARI OVASKA

Department of Electrical Engineering
Helsinki University of Technology
Otakaari 5 A, FI-02150 Espoo, Finland
Tel: +358 9 451 2434, Fax: +358 9 451 2432
http://powerelectronics.tkk.fi/
E-mail: {gao; xiaolei}@cc.hut.fi, seppo.ovaska@tkk.fi

Received June 2008; revised November 2008

ABSTRACT. *The Harmony Search (HS) method is an emerging meta-heuristic optimization algorithm. In this paper, we propose two modified HS methods to deal with the uni-modal and multi-modal optimization problems. The first modified HS method is based on the fusion of the HS and Differential Evolution (DE) technique, namely, HS-DE. The DE is employed here to optimize the members of the HS memory. The second modified HS method utilizes a novel HS memory management approach, and it targets at handling the multi-modal problems. Several nonlinear functions are used to demonstrate and verify the effectiveness of our two new HS methods.*

Keywords: Harmony search, differential evolution, uni-modal optimization, multi-modal optimization, hybrid optimization methods

1. Introduction. Firstly proposed by Geem *et al.* in 2001 [1], the HS method is inspired by the underlying principles of the musicians' improvisation of the harmony. During the recent years, it has been successfully applied in the areas of function optimization [2], mechanical structure design [3], and pipe network optimization [4]. Unfortunately, empirical study has shown that the original HS method sometimes suffers from a slow search speed [2], and it is not suitable for handling the multi-modal problems. To overcome these drawbacks, we propose two modified HS methods in this paper. The first modified HS method is a hybridization of the HS and Differential Evolution (DE): HS-DE, which can accelerate the convergence procedure of the regular HS method. The DE technique is a simple but universal numerical optimizer [5]. The individuals in the DE are updated by an amount of the difference between two randomly chosen ones. The DE has the distinguishing advantages of computation simplicity as well as convergence efficiency. The second modified HS method is based on the employment of an effective diversity maintenance policy for the members of the HS memory. Extensive computer simulations have shown that our two modified HS methods can outperform the original HS in attacking the uni-modal and multi-modal problems.

The rest of this paper is organized as follows. We briefly introduce the essential principles of both the HS and DE methods in Sections 2 and 3, respectively. In Section 4, by merging the HS and DE together, we propose a new hybrid optimization method: HS-DE, in which the fitness of the HS memory members can be improved by the DE. The second modified HS method is presented and discussed in Section 5. Simulation examples of nonlinear functions optimization are demonstrated in Section 6. Finally, in Section 7, we conclude our paper with some remarks and conclusions.

2. Harmony Search Method. As we know, when musicians compose the harmony, they usually try various possible combinations of the music pitches stored in their memory. This kind of efficient search for a perfect harmony is analogous to the procedure of finding the optimal solutions to engineering problems. The HS method is inspired by the working principles of the harmony improvisation [1]. Figure 1 shows the flowchart of the basic HS method, in which there are four principal steps involved.

Step 1. Initialize the HS Memory (HM). The initial HM consists of a given number of randomly generated solutions to the optimization problems under consideration. For an n-dimension problem, an HM with the size of N can be represented as follows:

$$\mathrm{HM} = \begin{bmatrix} x_1^1, x_2^1, \cdots, x_n^1 \\ x_1^2, x_2^2, \cdots, x_n^2 \\ \vdots \\ x_1^N, x_2^N, \cdots, x_n^N \end{bmatrix}, \tag{1}$$

where $\left[x_1^i, x_2^i, \cdots, x_n^i\right]$ ($i = 1, 2, \cdots, N$) is a solution candidate. N is typically set to be between 10 and 100.

Step 2. Improvise a new solution $\left[x_1', x_2', \cdots, x_n'\right]$ from the HM. Each component of this solution, x_j', is obtained based on the Harmony Memory Considering Rate (HMCR). The HMCR is defined as the probability of selecting a component from the present HM members, and 1-HMCR is, therefore, the probability of generating it randomly. If x_j' comes from the HM, it is chosen from the j^{th} dimension of a random HM member, and it can be further mutated according to the Pitching Adjust Rate (PAR). The PAR determines the probability of a candidate from the HM to be mutated. Obviously, the improvisation of

$[x'_1, x'_2, \cdots, x'_n]$ is rather similar to the production of the offspring in the Genetic Algorithms (GA) [6] with the mutation and crossover operations. However, the GA creates fresh chromosomes using only one (mutation) or two (simple crossover) existing ones, while the generation of new solutions in the HS method makes full use of all the HM members.

Step 3. Update the HM. The new solution from Step 2 is evaluated. If it yields a better fitness than that of the worst member in the HM, it will replace that one. Otherwise, it is eliminated.

Step 4. Repeat Step 2 to Step 3 until a preset termination criterion, e.g., the maximal number of iterations, is met.

Similar to the GA and particle swarm algorithms [7]-[9], the HS method is a random search technique. It does not require any prior domain knowledge, such as the gradient information of the objective functions. However, different from those population-based evolutionary approaches, it only utilizes a single search memory to evolve. Therefore, the HS method has the feature of algorithm simplicity. Note that the HS memory stores the past search experiences, and plays an important role in its optimization performance. In the next section, we employ the DE method to improve the fitness of all the members in the HS memory so that the overall convergence speed of the original HS method can be accelerated.

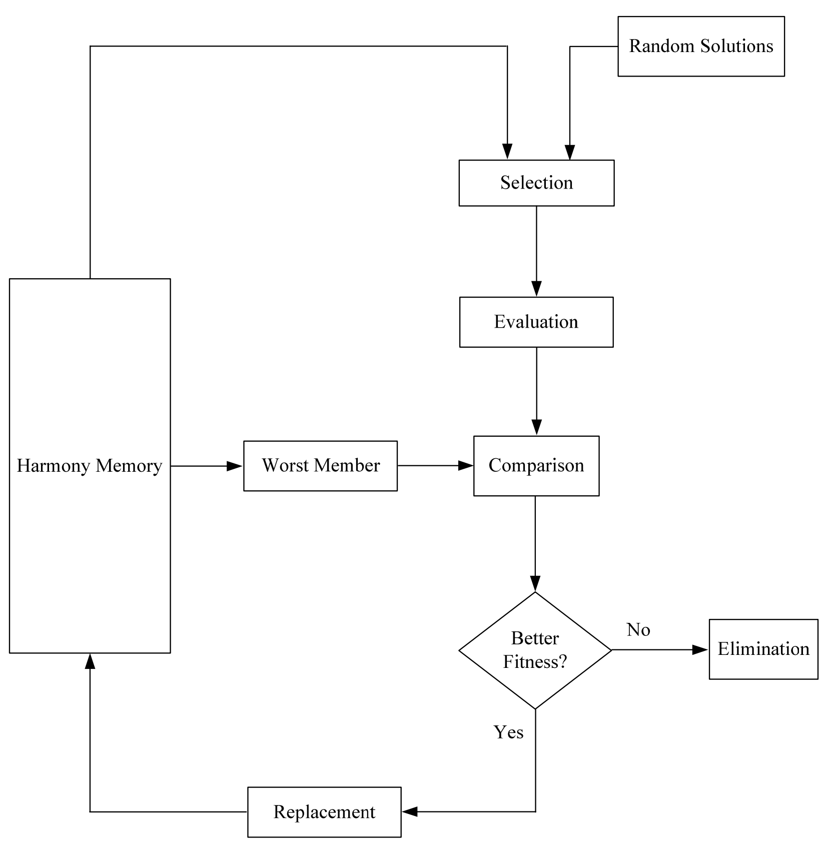


FIGURE 1. Harmony Search (HS) method

3. Modified HS Method for Uni-Modal Optimization. *A. Differential Evolution (DE) method.* The DE method is a robust population-based optimization technique firstly proposed by Storn and Price [5] [10]-[12]. The principle of the DE is similar to that of other evolutionary computation methods, such as the GA. However, the unique idea of the DE is that it generates new chromosomes by adding the weighted difference between two random chromosomes to the third one. If the fitness of the resulting chromosome is better than that chromosome, this newly generated chromosome replaces the one with which it is compared. The simplest DE can be explained as the following. Suppose there are three chromosomes, $r_1(k)$, $r_2(k)$, and $r_3(k)$, in the current population, as shown in Figure 2. A trial update of $r_3(k)$, $r_3'(k+1)$, is given:

$$r_3'(k+1) = \lambda_1 r_3(k) + \lambda_2 \left[r_1(k) - r_2(k) \right], \tag{2}$$

where λ_1 and λ_2 are two pre-determined weights. In order to further increase the diversity of the chromosomes, a 'crossover' operator is employed to generate $r_3''(k+1)$ by randomly combining those parameters of $r_3(k)$ and $r_3'(k)$ together. If $r_3''(k+1)$ yields a higher fitness than $r_3(k)$, we get:

$$r_3(k+1) = r_3''(k+1) \,. \tag{3}$$

Otherwise, $r_3''(k+1)$ is eliminated, and the above iteration procedure will restart. $r_1(k)$ and $r_2(k)$ are normally randomly selected from the population, and should be mutually different from each other. Apparently, the update of the chromosomes in the DE method is similar to the crossover operator of the GA. As a matter of fact, the difference between two chromosomes is an estimation of the gradient information in that zone, where both chromosomes belong to. Hence, the DE can be also considered as a gradient descent-based random search method.

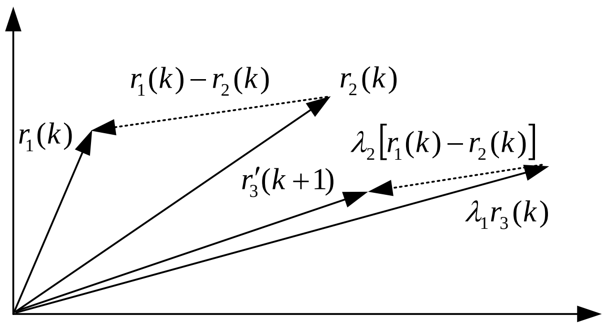


FIGURE 2. Differential Evolution (DE) method

During the past decade, hybridization of evolutionary computing algorithms has gained considerable popularity, which can overcome their individual drawbacks while benefit from each other's strengths [13]-[16]. As aforementioned, the DE method has the remarkable advantage of effective search. Thus, we propose a fusion of the HS and DE: HS-DE, which is capable of significantly outperforming the regular HS approach in the uni-modal optimization.

B. Fusion of HS and DE: HS-DE

It is well known that the HM storing the elite solutions acquired in the history has a central effect on the behavior of the HS method. Unfortunately, the update of the HS memory members is solely based on the past search experiences. In this section, we develop the HS-DE, in which the DE technique is applied to fine-tune the HM. More precisely, all the members of the HM are regarded as the DE individuals, and they can evolve together in the population of the DE. For example, $[x_1', x_2', \cdots, x_n']$ is updated to $[x_1'', x_2'', \cdots, x_n'']$ after a given number of the DE iterations. Every element in the HM needs to go through the above DE-based refinement procedure. Hence, the resulting HM members are expected to have better fitness than that of the original ones. Obviously, the updated HM can provide an improved basis for the HS method.

The proposed HS-DE has three interesting features. Firstly, the DE technique used takes full advantage of the information sharing and exchange among the members of the HM. This strategy can overcome the premature shortcoming of the regular HS method. Secondly, the DE-based update of the HM runs independently and in parallel with the search of the HS method. The DE is actually embedded into the HS method as a separate fine-tuning unit. Thirdly, the employment of the DE only moderately increases the computational complexity of the HS method. In Section 5, we demonstrate that this HS-DE has a superior optimization performance over the original HS method in coping with the uni-modal problems.

4. Modified HS Method for Multi-Modal Optimization. Optimization is referred to a process of finding the best solution or operating a system in the most effective way, particularly under some given constraints [17] [18]. Multi-modal optimization is an important but challenging topic in the field of optimization [19]-[21]. Unfortunately, it is difficult for the HS method to locate all the global optima of the multi-modal problems, because the HM members can be easily stagnated into one or several of them during iteration. Thus, the key issue of applying the HS method for the multi-modal optimization is to effectively maintain the diversity of the HM members. The regular HM management policy is explained in Section 2. However, some additional approaches are needed to determine whether a solution from Step 2 can replace the worst member in the HM. Indeed, the qualification of a solution candidate as a new HM member should be based on not only its fitness but also its similarity to all the existing members.

Inspired by the artificial fish swarm algorithm [22], we propose a new control mechanism, as shown in Figure 3, for updating the HM in our second modified HS method so as to attack the multi-modal problems. Suppose the fitness of the current HM members is denoted as f_i ($i = 1,2,\cdots,N$). After a solution candidate, $[x'_1, x'_2, \cdots, x'_n]$, with the fitness, f', is obtained, we first measure its distances, d_i ($i = 1,2,\cdots,N$), to all the HM members:

$$d_i = \left\| [x'_1, x'_2, \cdots, x'_n] - [x^i_1, x^i_2, \cdots, x^i_n] \right\|, \tag{4}$$

where $\| \ \|$ is an appropriately selected distance metric. Next, we calculate the number of the HM members, M, which are in the vicinity, V, of $[x'_1, x'_2, \cdots, x'_n]$. In other words, only the HM members, whose d_i are smaller than V, are counted here. The average fitness of these 'nearby' HM members, $\overline{F}$, is given as follows:

$$\overline{F} = \frac{\sum_{i=1}^{M} f_i}{M}. \tag{5}$$

Therefore, $[x'_1, x'_2, \cdots, x'_n]$ will replace the worst member of the HM, if it meets the following three conditions:

1. f' is greater than that of the worst HM member,
2. M is smaller than a preset threshold M_V,
3. f' is greater than $\overline{F}$.

It is concluded from the above explanations that our approach can prevent the harmful over-similarity among the HM members so that the diversity of the HS solutions is maintained. That is to say, the modified HS method is well-suited for handing the multi-modal problems. Nevertheless, the proposed technique has two drawbacks. Firstly, the parameters V and M_V are always applications dependent, and are usually chosen based on *trial and error*. They can significantly affect the multi-modal optimization performance of the modified HS method. Unfortunately, there is no analytic way yet to guarantee their best values. Secondly, as in (4), the distances between $[x_1', x_2', \cdots, x_n']$ and all the present HM members have to be calculated. This requirement can certainly result in a time-consuming procedure in case of a large N.

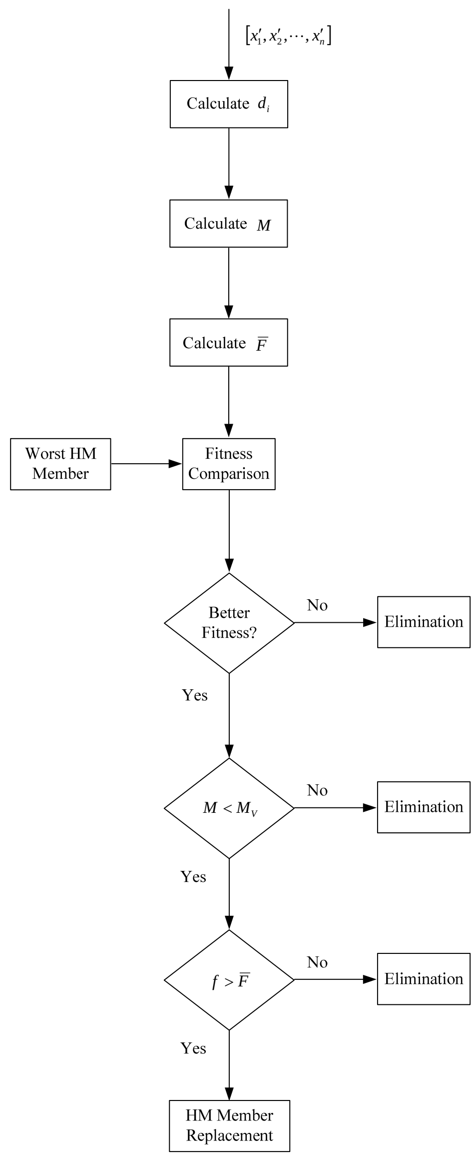


FIGURE 3. HM member control in modified HS method for multi-modal optimization

5. Simulations. In this section, we investigate the effectiveness of the two modified HS methods with a few simulation examples of uni-modal and multi-modal functions.

A. Uni-modal functions optimization

The following eleven *n*-dimension nonlinear functions, which have been widely used as the optimization benchmarks [23] [24], are employed to compare the optimization (minimization) capabilities between the HS and our HS-DE. Here, $n = 50$, except for Powell function, where $n = 52$.

Ackley function:

$$\mathrm{f}(x) = -20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)} + 20 - e, \quad x \in [-32,32]. \tag{6}$$

Dixon and Price function:

$$\mathrm{f}(x) = (x_1+1)^2 + \sum_{i=1}^{n} i(2x_i^2 - x_{i-1})^2, \quad x \in [-10,10]. \tag{7}$$

Levy function:

$$\mathrm{f}(x) = \sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2\left[1+\sin^2(3\pi x_{i+1})\right] + (x_n - 1)\left[1+\sin^2(3\pi x_n)\right], \quad x \in [-10,10]. \tag{8}$$

Michalewicz function:

$$\mathrm{f}(x) = -\sum_{i=1}^{n}\sin(x_i)\left[\sin\left(\frac{ix_i^2}{\pi}\right)\right]^{20}, \quad x \in [0,\pi]. \tag{9}$$

Perm function:

$$\mathrm{f}(x) = \sum_{k=1}^{n}\left\{\sum_{i=1}^{n}(i^k + 0.5)\left[\left(\frac{x_i}{i}\right)^k - 1\right]\right\}^2, \quad x \in [-n,n]. \tag{10}$$

Powell function:

$$\mathrm{f}(x) = \sum_{i=1}^{\frac{n}{4}}(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^2, \; x \in [-4,5]. \tag{11}$$

Rastrigin function:

$$\mathrm{f}(x) = \sum_{i=1}^{n} x_i^2 + 10 - 10\cos(2\pi x_i), \quad x \in [-5.12,5.12]. \tag{12}$$

Rosenbrock function:

$$\mathrm{f}(x) = \sum_{i=1}^{n} 100\left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2, \quad x \in [-10,10]. \tag{13}$$

Sphere function:

$$f(x)=\sum_{i=1}^{n} x_i^2, \quad x\in[-100,100]. \tag{14}$$

Trid function:

$$f(x)=\sum_{i=1}^{n}(x_i-1)^2-\sum_{i=2}^{n}x_i x_{i-1}, \quad x\in\left[-n^2,n^2\right]. \tag{15}$$

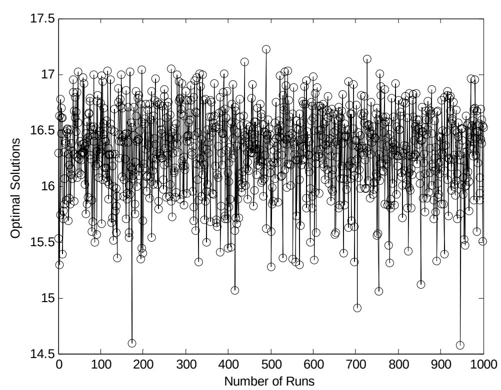
Zakharov function:

$$f(x)=\sum_{i=1}^{n} x_i^2+\left(\sum_{i=1}^{n}\frac{ix_i}{2}\right)^2+\left(\sum_{i=1}^{n}\frac{ix_i}{2}\right)^4, \quad x\in[-5,10]. \tag{16}$$

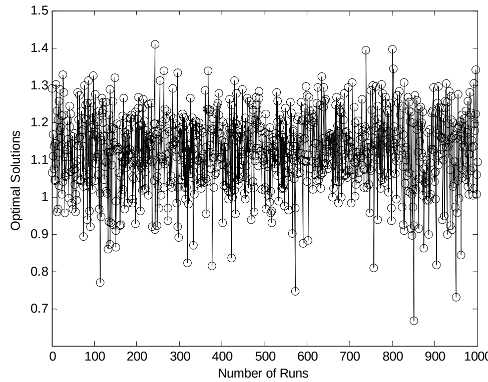
The global minima of all the above functions are at $f(x)=0$, except for Michalewics function and Trid function, whose global minima are unknown when $n=50$. Generally, evaluation of the objective function is the most time consuming part of nearly all the optimization algorithms. Therefore, we use the Number of Function Evaluation (NFE) rather than number of iterations as the principal criterion to compare the convergence speeds of the HS and HS-DE. Both of them have 100 HM members, i.e., $N=100$, which are initialized to be equal. The relevant parameters in these two methods are as follows: $\text{HMCR}=0.8$, $\text{PAR}=0.6$, $\lambda_1=0.7$, and $\lambda_2=0.3$. Their evolution procedures are terminated after 10,000 NFE. Table 1 gives the optimal solutions acquired. We stress that the results here are based on the average of 1,000 independent trials. As two illustrative examples, the optimal solutions to the Ackley function and Rastrigin function from the HS and HS-DE are shown in Figures. 4 and 5, respectively. Apparently, compared with the original HS method, for the eleven test functions, our HS-DE can achieve much better optimization results within the same NFE, due to the efficient DE-based refinement of the HM members. That is, the HS-DE has a superior uni-modal optimization capability over the HS method. However, the optimization effectiveness of this HS-DE can significantly deteriorate with inappropriately chosen parameters λ_1 and λ_2. Figures 6 (a) and (b) illustrate the optimal solutions to Rosenbrock function obtained by the HS-DE with $\lambda_1=0.7$ & $\lambda_2=0.3$ and $\lambda_1=0.9$ & $\lambda_2=0.1$, respectively. We can observe that λ_1 and λ_2 indeed play a pivotal role in our HS-DE.

TABLE 1. Optimal solutions acquired by HS and HS-DE within 10,000 NFE

	HS	HS-DE
Ackley Function	16.3014	1.1232
Dixon and Price Function	3.9617×10^{5}	0.9345
Levy Function	63.8282	4.2580
Michalewicz Function	- 29.5973	-13.7792
Perm Function	1.6919×10^{14}	1.5122×10^{14}
Powell Function	5.9420×10^{3}	6.5404×10^{-8}
Rastrigin Function	273.5232	16.8533
Rosenbrock Function	4.7523×10^{5}	54.7093
Sphere Function	2.1569×10^{4}	5.7988
Trid Function	1.6403×10^{7}	26.8870
Zakharov Function	1.2005×10^{3}	77.8902

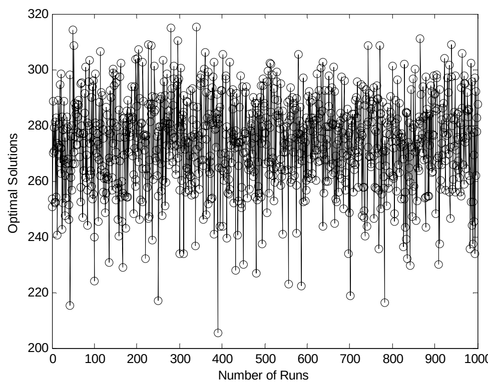


(a)

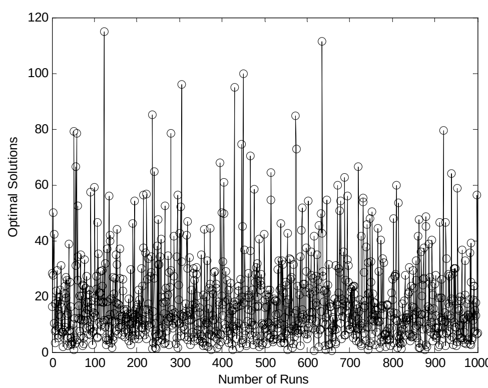


(b)

FIGURE 4. Optimal solutions to Ackley function acquired by HS and HS-DE
(a) HS (b) HS-DE

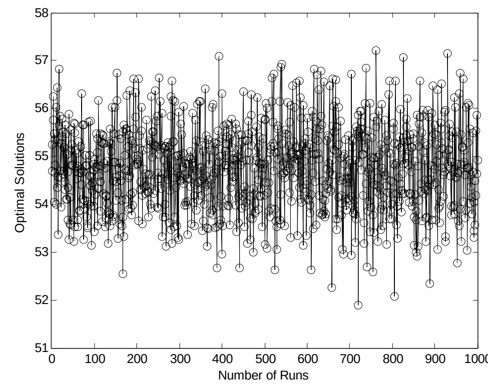


(a)

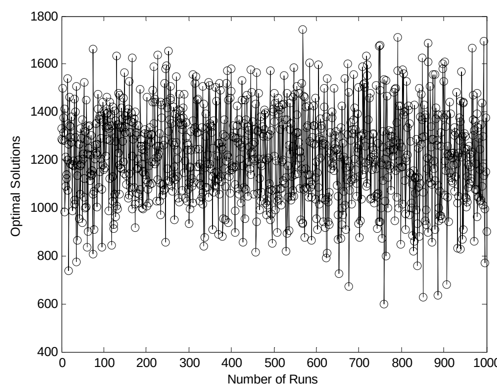


(b)

FIGURE 5. Optimal solutions to Rastrigin function acquired by HS and HS-DE
(a) HS (b) HS-DE



(a)



(b)

FIGURE 6. Optimal solutions to Rosenbrock function acquired by HS-DE with DE different parameters

(a) $\lambda_1 = 0.7$ and $\lambda_2 = 0.3$ (b) $\lambda_1 = 0.9$ and $\lambda_2 = 0.1$

B. Multi-modal functions optimization

In this example, the multi-modal optimization capability of the second modified HS method is examined using the following three two-dimension functions [25] [26]:

$$\mathrm{f}_1(x, y) = 200 - (x^2 + y - 11)^2 + (x + y^2 - 7)^2, \quad -5 \le x, y \le 5. \tag{17}$$

$$\mathrm{f}_2(x, y) = x\sin(4\pi x) - y\sin(4\pi y + \pi) + 1, \quad -1 \le x, y \le 1. \tag{18}$$

$$\mathrm{f}_3(x, y) = \left(\frac{3}{0.5 + x^2 + y^2}\right)^2 + \left(x^2 + y^2\right)^2, \quad -5.12 \le x, y \le 5.12. \tag{19}$$

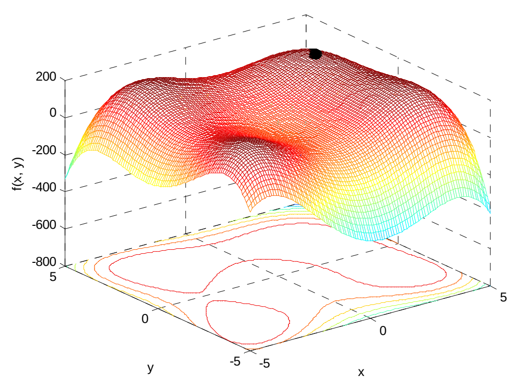
Each function has only one global optimum (minimum) but several local optima. Actually, the goal of the optimization algorithms employed is to find not only the global optimum but also as many local optima as possible. The typical optimization results of these functions of the regular HS and modified HS methods after 100,000 iterations are illustrated in Figures. 7-9. $N = 100$, $\text{HMCR} = 0.75$, and $\text{PAR} = 0.6$ are used in both two optimization techniques. In our modified HS method, V and M_V for $f_1(x, y)$, $f_2(x, y)$, and $f_3(x, y)$ are given as follows:

For $f_1(x, y)$, $V = 0.15$, and $M_V = 2$.

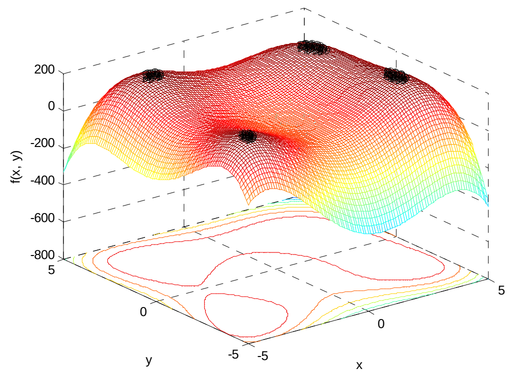
For $f_2(x, y)$, $V = 0.075$, and $M_V = 3$.

For $f_3(x, y)$, $V = 0.025$, and $M_V = 3$.

It is clearly visible that the regular HS method can only find the global optimum of each function, while the modified HS method is capable of locating most of the local optima in addition to the global one. Nevertheless, we emphasize that like λ_1 and λ_2 in the HS-DE, V and M_V can also considerably affect the multi-modal optimization performance of our modified HS method. For example, Figures 10 (a) and (b) show the optimization results of $f_2(x, y)$, when $V = 0.01$ and $V = 1$, respectively, which are apparently worse than that in Figure 8 (b). In case of a fixed M_V, if V is too small, the behaviors of the normal HS and modified HS methods are quite similar. On the other hand, with a too large V, the quality of the global and local optima located indeed become poor. Unfortunately, how to choose the best V and M_V is still an unsolved problem, although some adaptation strategies can be the potential solutions.



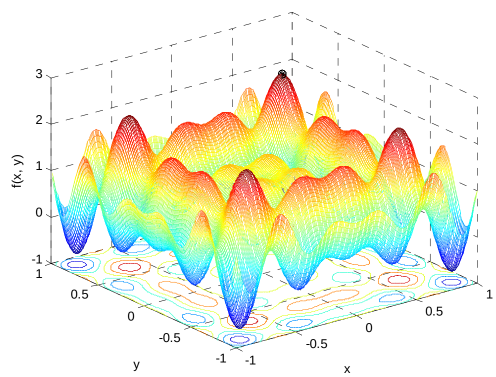
(a)



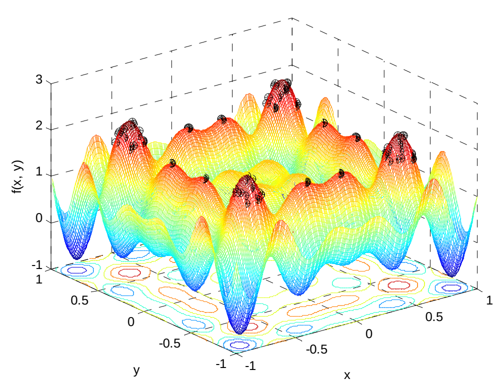
(b)

FIGURE 7. Optimization results of $f_1(x, y)$ using regular HS and modified HS methods

(a) regular HS method (b) modified HS method



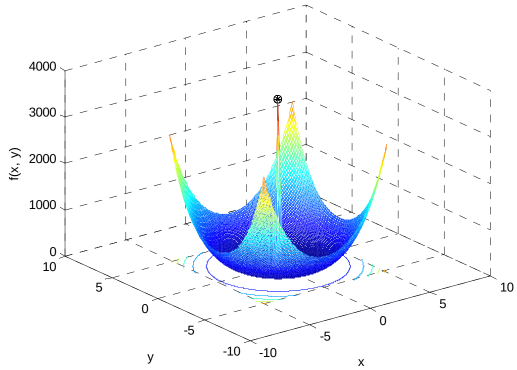
(a)



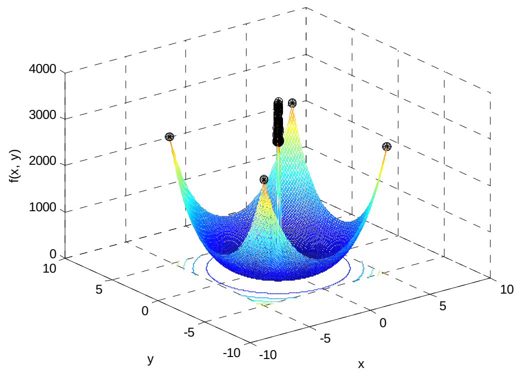
(b)

FIGURE 8. Optimization results of $f_2(x, y)$ using regular HS and modified HS methods

(a) regular HS method (b) modified HS method



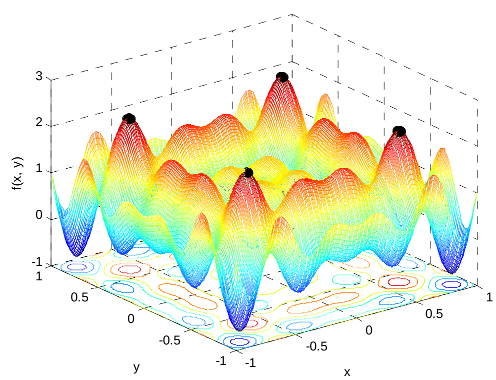
(a)



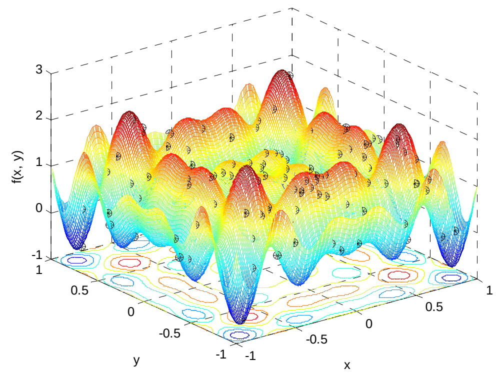
(b)

FIGURE 9. Optimization results of $f_3(x, y)$ using regular HS and modified HS methods

(a) regular HS method (b) modified HS method



(a)



(b)

FIGURE 10. Optimization results of $f_2(x, y)$ using modified HS method with $V = 0.01$ and $V = 1$ (a) $V = 0.01$ (b) $V = 1$

6. Conclusions. In this paper, we propose two modified HS methods to deal with the uni-modal and multi-modal problems. Based on the fusion of the HS and DE, a novel hybrid optimization scheme, HS-DE, is first discussed. The HM members are fine-tuned by the DE to improve their affinities so that enhanced optimization performances can be achieved. In the second modified HS method, we employ a fish swarm-based technique to maintain the diversity of the HM members, which makes it a suitable candidate for handling the multi-modal problems. Several simulation examples of the uni-modal and multi-modal functions have been used to verify the effectiveness of the proposed methods. Compared with the original HS, better optimization results are obtained using our modified HS approaches. However, a few important issues, such as convergence analysis and optimal parameters selection, need to be further explored. We are also going to study how to apply these modified HS methods in handling real-world problems.

Acknowledgments. This research work was funded by the Academy of Finland under Grant 214144. The authors would like to thank the anonymous reviewers for their insightful comments and constructive suggestions that have improved the paper.

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] K. S. Lee and Z. W. Geem, A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, *Computer Methods in Applied Mechanics and Engineering*, vol.194, no.36-38, pp.3902-3922, 2005.
[3] S. L. Kang and Z. W. Geem, A new structural optimization method based on the harmony search algorithm, *Computers and Structures*, vol.82, no.9-10, pp.781-798, 2004.

[4] Z. W. Geem, J. H. Kim and G. V. Loganathan, Harmony search optimization: application to pipe network design, *International Journal of Modeling and Simulation*, vol.22, no.2, pp.125-133, 2002.
[5] R. Storn and K. Price, Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces, *Journal of Global Optimization*, vol.11, no.4, pp.341-359, 1997.
[6] R. Poli and W. B. Langdon, *Foundations of Genetic Programming*, Berlin, Germany: Springer-Verlag, 2002.
[7] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, West Sussex, England: John Wiley & Sons Ltd, 2005.
[8] X. Cai, Z. Cui, J. Zeng and Y. Tan, Particle swarm optimization with self-adjusting cognitive selection strategy, *International Journal of Innovative Computing, Information and Control*, vol.4, no.4, pp.943-952, 2008.
[9] L. Deng, Y. Lin, W. Zheng and Y. Xi, Incorporating justification in the particle swarm optimization for the RCPSP, *International Journal of Innovative Computing, Information and Control*, vol.4, no.9, pp. 2315-2324, 2008.
[10] M. Toman, G. Stumberger and D. Dolinar, Parameter identification of the Jiles-Atherton hysteresis model using differential evolution, *IEEE Transactions on Magnetics*, vol.44, no.6, pp.1098-1101, 2008.
[11] G. Y. Yang, Z. Y. Dong and K. P. Wong, A modified differential evolution algorithm with fitness sharing for power system planning, *IEEE Transactions on Power Systems*, vol.23, no.2, pp.514-522, 2008.
[12] S. Das, A. Abraham and A. Konar, Automatic clustering using an improved differential evolution algorithm, *IEEE Transactions on Systems, Man, and Cybernetics-Part A*, vol.38, no.1, pp.218-237, 2008.
[13] C. Grosan, A. Abraham and H. Ishibuchi (Eds.), *Hybrid Evolutionary Algorithms*, Berlin, Germany: Springer-Verlag, 2007.
[14] X. Wang, X. Z. Gao and S. J. Ovaska, A hybrid optimization algorithm based on ant colony and immune principles, *International Journal of Computer Science and Applications*, vol.4, no.3, pp.30-44, 2007.
[15] X. Wang, X. Z. Gao, and S. J. Ovaska, A simulated annealing-based immune optimization method, *Proc. of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*, Porvoo, Finland, September 2008, pp.41-47.
[16] X. Wang, X. Z. Gao and S. J. Ovaska, A novel particle swarm-based method for nonlinear function optimization, *International Journal of Computational Intelligence Research*, vol.4, no.3, 2008, in Press.
[17] Y. Takeuchi, Optimization of linear observations for the stationary Kalman filter based on a generalized water filling theorem, *International Journal of Innovative Computing, Information and Control*, vol.4, no.1, pp.211-230, 2008.
[18] T. Furusho, T. Nishi and M. Konishi, Distributed optimization method for simultaneous

production scheduling and transportation routing in semiconductor fabrication bays, *International Journal of Innovative Computing, Information and Control*, vol.4, no.3, pp.559-578, 2008.

[19] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Reading, MA: Addison-Wesley, 1989.

[20] X. Xu and J. Zhang, An improved immune evolutionary algorithm for multimodal function optimization, *Proc. of the 3rd International Conference on Natural Computation*, Haikou, China, August 2007, pp.641-646.

[21] T. Tang and J. Qiu, An improved multimodal artificial immune algorithm and its convergence analysis, *Proc. of the 6th World Congress on Intelligent Control and Automation*, Dalian, China, June 2006, pp.3335-3339.

[22] C.-R. Wang, C.-L. Zhou and J.-W. Ma, An improved artificial fish-swarm algorithm and its application in feed-forward neural networks, *Proc. of the* 2005 *International Conference on Machine Learning and Cybernetics*, Guangzhou, China, August 2005, pp.2890-2894.

[23] Benchmark functions are available at http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm.

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

[25] L. N. de Castro and F. J. von Zuben, Learning and optimization using the clonal selection principle, *IEEE Transactions on Evolutionary Computation*, vol.6, no.3, pp.239-251, 2002.

[26] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd edition, Berlin, Germany: Springer-Verlag, 1996.