Gao, Xiaozhi; Wang, Xiaolei; Zenger, Kai

Harmony search method for optimal wind turbine electrical generator design

Published in:
Rakenteiden Mekaniikka (Journal of Structural Mechanics)

Published: 30/12/2016

Document Version
Publisher's PDF, also known as Version of record

Rakenteiden Mekaniikka (Journal of Structural Mechanics)
Vol. 49, No 3, 2016, pp. 119-136
rmseura.tkk.fi/rmlehti/

VERTAISARVIOITU
KOLLEGIALT GRANSKAD
PEER-REVIEWED
www.tsv.fi/tunnus

Harmony search method for optimal wind turbine electrical generator design

Xiao-Zhi Gao, Xiaolei Wang, and Kai Zenger

Summary. The Harmony Search (HS) method is an emerging meta-heuristic optimization algorithm, which has been employed to cope with numerous challenging tasks during the past decade. In this paper, the essential theory of the HS algorithm is first described in details. Next, a few typical variations of the HS method are explained. The application of the HS in a practical wind turbine electrical generator optimal design case study is finally presented. Computer simulation results have clearly demonstrated its remarkable performances in dealing with demanding optimization problems.

Key words: nature-inspired computing methods, Harmony Search (HS) method, Population-Based Incremental Learning (PBIL), optimization, electrical machine design

Received 20 November 2015. Accepted 8 June 2016. Published online 30 December 2016.

Introduction

Firstly, proposed by Geem *et al.* in 2001 [10], the Harmony Search (HS) method is inspired by the underlying principles of the musicians' improvisation of the harmony. The HS has the distinguishing features of algorithm simplicity and search efficiency. During the recent years, it has been successfully used in such areas as function optimization [19], mechanical structure design [18], pipe network optimization [11], optimization of data classification systems [26], and stochastic equilibrium network design [1]. In our paper, we first introduce the underlying inspiration and principles of the basic HS method in Section *Harmony search method*. Some representative modified HS algorithms are next explained in Section *Representative variants of HS method*. The application of the HS in a case study of wind turbine electrical generator design optimization is further presented in Section *Application of HS method in optimal wind turbine electrical generator design*. Finally, in Section *Conclusions*, this paper is concluded with some remarks and conclusions.

Harmony search method

When musicians compose the harmony, they usually try various possible combinations of the music pitches stored in their memory. This search for the perfect harmony is indeed analogous to the procedure of finding the optimal solutions to engineering problems. The HS method is actually inspired by the working principles of the harmony improvisation [10]. Figure 1 shows the flowchart of the basic HS method, in which there are four principal steps involved. The pseudo code of the HS is given in Figure 2.

Step 1. Initialize the HS Memory (HM). The initial HM consists of a certain 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:

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

where $\left[x_1^i, x_2^i, \cdots, x_n^i\right]$ ($i = 1,2,\cdots,HMS$) is a solution candidate. HMS is typically set to be between 50 and 100. As a matter of fact, this representative choice of the HM size in the HS method is based on the empirical study reported in intensive literature.

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 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 is further mutated according to the Pitching Adjust Rate (PAR). The PAR determines the probability of a candidate from the HM to be mutated. As we can see, the improvisation of $\left[x_1', x_2', \cdots, x_n'\right]$ is rather similar to the production of offspring in the Genetic Algorithms (GA) [24] with the mutation and crossover operations. However, the GA creates new 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.

Note that in Step 4, there are some other termination criteria for stopping the HS method. For example, if the solution obtained is better than a pre-set value, the running of the HS method can be terminated. Another commonly used termination criterion is that the HS iteration procedure is stopped, when no significant improvement in the solutions acquired can be observed during the consecutive iterations. Similar to the GA and swarm intelligence algorithms [4] [16], 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 approaches,

it only utilizes a single search memory to evolve. Therefore, the HS method has the distinguishing feature of computational simplicity.

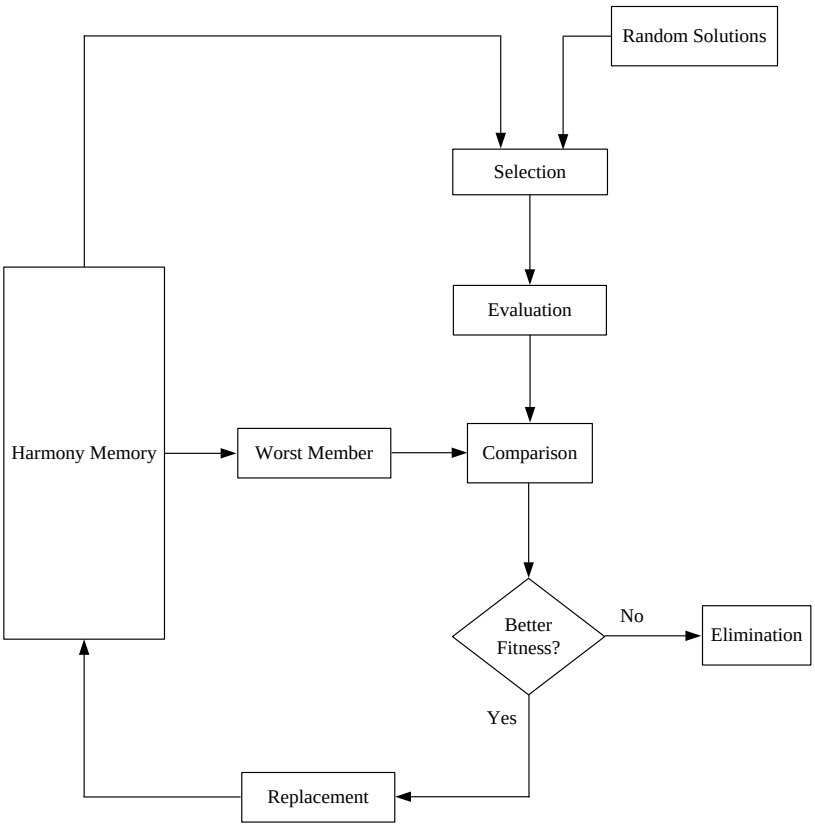

Figure 1. Harmony Search (HS) method.

/* HM initialization */
for (*i*=1; *i*<= *HMS*; *i*++)
  for (*j*=1; *j*<=n; *j*++)
    Randomly initialize x_j^i in HM.
  endfor
endfor
/* End of HM initialization */
Repeat
  /* Construction and evaluation of new solution candidate **x** */
  for (*j*=1; *j*<=n; *j*++)
    if (rand(0, 1)<HMCR)
      Let x_j in **x** be the *j* th dimension of a randomly selected HM member.
      if (rand(0, 1)<PAR)

Apply pitch adjustment distance *bw* to mutate x_j:

$x_j = x_j \pm \text{rand}(0,1) \times bw$.

endif

else

Let x_j in $\mathbf{x}$ be a random value.

endif

endfor

Evaluate the fitness of $\mathbf{x}$: $f(\mathbf{x})$.

/* End of construction and evaluation of new solution candidate $\mathbf{x}$ */

/* HM update */

if ($f(\mathbf{x})$ is better than the fitness of the worst HM member)

Replace the worst HM member with $\mathbf{x}$.

else

Disregard $\mathbf{x}$.

endif

/* End of HM update */

Until a preset termination criterion is met.

Figure 2. Pseudo code of HS method.

Representative variants of HS method

A lot of modified HS algorithms have been studied in the past decade so as to enhance the optimization performances of the original version. As a matter of fact, a special discrete variation of the HS is proposed by Geem on the basis of introducing the stochastic derivatives for the discrete variables involved [12]. The stochastic derivatives give the selection probabilities of certain discrete variables during the evolution procedure of the HS. It is efficient at manipulating discrete optimization problems, and has been employed in the optimal design of fluid-transport networks. Omran and Mahdavi embed the ideas borrowed from swarm intelligence into the regular HS, and develop a new variant: Global-best HS (GHS) [22]. In the GHS, the adjustment of new solutions improvised is only based on the best harmony selected from the HM without the involvement of the distance bandwidth (*bw*). This interesting approach adds the unique social learning capability to the GHS. The investigation experiments of ten benchmark functions prove that the GHS can generally outperform the original HS. Inspired by the local versions of the Particle Swarm Optimization (PSO) [20] and GHS, Pan *et al*. propose a local-best variant of the HS method with dynamic subpopulation: DLHS [23].

In the DLHS, the whole HM is divided into multiple sub-HMs, each of which can evolve independently. However, these sub-HMs will form the HM again after searching for the optimal solutions in their own regions. With this subpopulation policy and a simple local search strategy, the DLHS is capable of achieving a satisfactory compromise between the exploration and exploitation in search. It has been successfully applied to attack the difficult lot-streaming flow shop scheduling problem. Inspired by the GHS and DLHS, Geem further develops the Particle-Swarm Harmony Search (PSHS) [13], which has been validated to be better than the original HS algorithm for small-size problems, but worse in case of large-scale one. A few novel hybrid HS methods have also been introduced by the authors of the present paper [5–9, 27].

The parameters of HMCR and PAR usually play a critical role in the optimization performance of the HS method. Unfortunately, properly choosing the appropriate values for them is always a challenging topic. Mahdavi *et al*. study an adaptive strategy for adjusting PAR and *bw* in the Improved HS (IHS) algorithm [21]. The values of PAR and *bw* dynamically increase and decrease with the growth of HS iterations, respectively so as to enhance the performance of the IHS. The IHS has been demonstrated to achieve comparable performances with other evolutionary and mathematical programming techniques in dealing with several test problems in terms of both the number of the fitness function evaluations required and quality of the solutions found. Unfortunately, the lower and upper limits for the update of PAR and *bw* are often case dependent, and are therefore difficult to determine. In [17], another adaptive HS method is proposed and explored. It takes advantage of two varying control parameters, η and ρ, to generate new harmony vectors. Both of these parameters are selected from the average values that are observed within the current harmony memory matrix using a given probability density function. This adaptive HS algorithm has found great successes in handling large steel structure optimization problems. Wang and Huang propose a new self-adaptive HS technique in [28]. Their almost parameter-free HS uses the information stored in the HM (self-consciousness), i.e., the minimum and maximum of the present HM members, to automatically control the pitch adjustment step. The low-discrepancy sequences are also utilized to initialize the HM. It has been compared with the aforementioned IHS and GHS, and can offer superior performances on four optimization problems tested. Geem and Sim introduce the Parameter-Setting-Free (PSF) technique to eliminate the common difficulty of selecting suitable HS parameters [14]. The developed PSF-HS has a new operation step, namely, rehearsal, in which certain number of new solutions is generated with the initial HMCR and PAR. The adaptive HMCR and PAR are then calculated based on the rehearsal results evaluated. The PSF-HS has been shown to be more robust than the original HS method, although its computational complexity is moderately high. The authors of the present paper also study a fusion of the HS and Cultural Algorithm (CA), HS-CA, in which the search knowledge stored in the CA is utilized to guide the mutation direction and size of the HS. This HS-CA is further used to effectively cope with an optimal wind turbine electrical generator design problem [7]. In [8], a hybrid HS method inspired by the opposition-based learning is proposed by the same authors. The HS method is merged with the Population-Based Incremental Learning (PBIL) for the optimal design of electrical machines in [9].

Theoretical research on the working principles and search mechanism of the HS method has also been reported in the recent literature, which can provide a useful guideline for users to design this algorithm in practice. Das *et al.* discuss the exploratory power of the HS by analyzing the evolution of the population variance over successive generations of the HM [3]. Based on their analysis, they further propose a modified HS algorithm, Exploratory HS (EHS), in which *bw* for the pitch adjustment is set to be proportional to the standard deviation of the HM population. In the simulation study, the EHS can not only outperform three existing HS variants, IHS, GHS, and MHS [2], over all the test functions but also yield better or at least comparable results when compared with a few state-of-the-art swarm intelligence techniques. Unfortunately, how to choose the optimal proportional gain *k* for *bw* in the EHS is still an open issue.

Application of HS method in optimal wind turbine electrical generator design

In the real world, modern science and industry are indeed rich in the problems of optimization. The HS has been originally proposed by Geem and applied to solve the optimization problem of water distribution networks in 2000 and since then the applications of the HS have covered numerous areas including industry, optimization benchmarks, power systems, medical science, control systems, construction design, and information technology [15]. In this section, we investigate and demonstrate the optimization effectiveness of the modified HS method in a real-world design problem: a wind turbine electrical generator design optimization.

Optimal design of a wind turbine electrical generator

Wind turbine electrical generator design is an important but demanding topic in the electrical machinery industry, due to, e.g., the number of optimization parameters. Generally speaking, the design of wind turbine electrical generators may involve more design parameters and stronger nonlinearity than other kinds of generators. The wind turbine electrical generator shown in Figure 3 is a radial flux type permanent magnet generator, in which the NdFeB magnets are surface mounted [25]. The remanence flux density of the magnets is 1.05 T, and coercivity 800 kA/m. The stator winding is a three-phase two-layer full-pitch diamond winding. The number of the slots per pole and phase is 2. The stator slot and constant dimensions of the slot are illustrated in Figure 4. The iron core consists of 55 mm long sub-cores, between which there are radial 6 mm wide ventilation ducts. The length of the sub-core is constant, and the number of the ventilation ducts is a decimal fraction in the calculations. The stator frame, bearing shields, and rotor steel body are all 20 mm thick. In the rotor body disc, there are holes, and around 50 % of the disc is iron and 50 % holes. The iron loss factor is p_{15} = 6.6 W/kg with 50 Hz and 1.5 T, and the air-gap length is 5 mm. The rated values of this practical wind turbine electrical generator are given in Table 1.

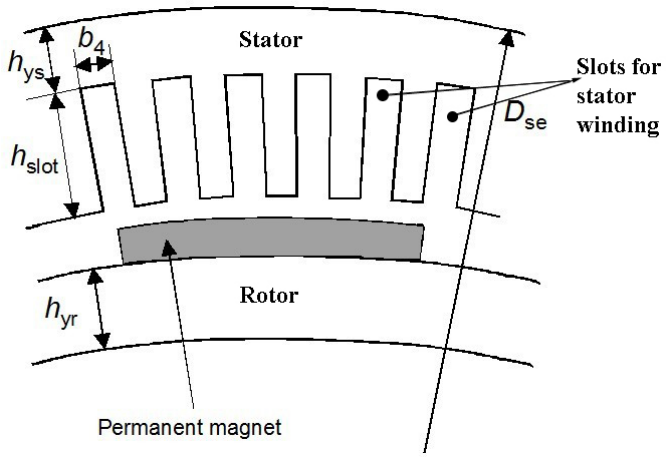

Figure 3. Cross-section and dimensions of permanent magnet generator.

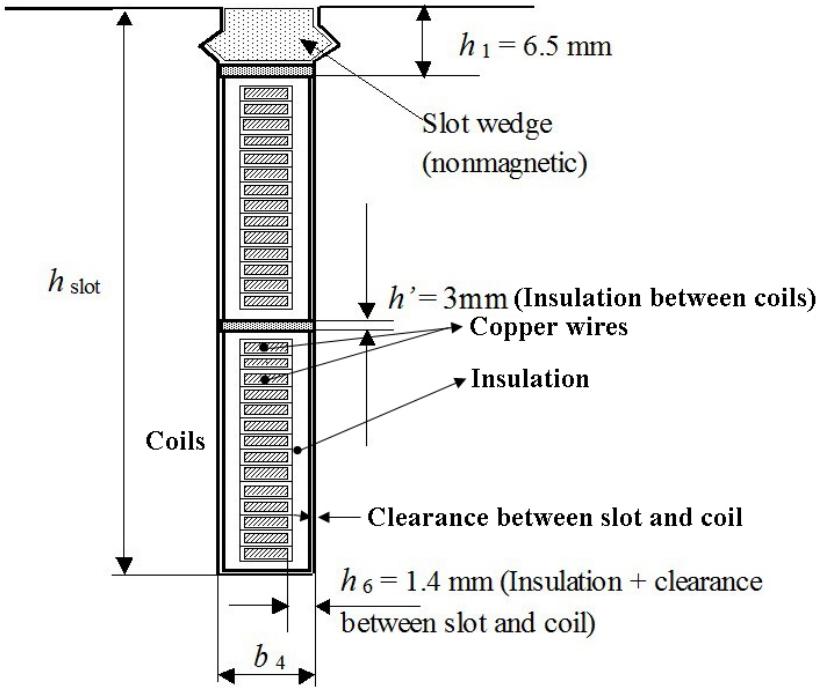

Figure 4. Slot form and constant dimensions of slot.

Table 1. Rate values of wind turbine electrical generator.

Parameter	Value
Power	3 MW
Voltage	690 V
Connection	Star
Speed	16.98 rpm
Number of phases	3

The detailed design principles of the above wind turbine electrical generator are explained in [25]. The objective function f($\boldsymbol{x}$) (in €) to be minimized is the sum of the material costs and capitalized costs of the total losses of this generator:

$$\mathrm{f}(\boldsymbol{x}) = k_{\mathrm{Fe}} m_{\mathrm{Fe}} + k_{\mathrm{Cu}} m_{\mathrm{Cu}} + k_{\mathrm{PM}} m_{\mathrm{PM}} + k_{\mathrm{Fef}} m_{\mathrm{Frame}} + k_{\mathrm{Loss}} P_{\mathrm{tot}}\,, \tag{2}$$

where m_{Fe}, m_{Cu}, m_{PM}, and m_{Frame} are the masses of the stator iron core, stator winding, permanent magnets, and stator frame and rotor body, respectively, k_{Fe}, k_{Cu}, k_{PM}, and k_{Fef} the unit prices of the stator core, copper, permanent magnets, and stator frame and rotor body, respectively, and k_{Loss} capitalized costs of the losses (Table 2). More details of the calculation of (2) can be found in [25]. The cost of the stator core actually includes the punching, waste parts of the sheet, as well as assembly of the stator core. The manufacturing cost of the winding is taken into account in the copper cost. The permanent magnet cost includes the corrosion protection, assembly into bigger cassettes, and magnetization of the magnets. The stator frame and rotor body costs consist of the material cost and cost of manufacturing the frame and body.

Table 2. Unit prices of materials and capitalized loss costs.

Unit	Price
Electrical steel, k_{Fe}	4 €/kg
Copper, k_{Cu}	12 €/kg
NdFeB magnets, k_{PM}	60 €/kg
Stator frame and rotor steel body, k_{Fef}	2 €/kg
Losses, k_{Loss}	2 €/W

The stator resistive losses are calculated at the temperature of 100 ^{0}C, and the iron losses in the stator teeth are

$$P_{\mathrm{Fed}} = 2 \cdot p_{15} \left(B_d / 1.5\mathrm{T}\right)^2 \left(f / 50\mathrm{Hz}\right)^{1.5} m_d\,, \tag{3}$$

where p_{15} is the iron loss factor, B_{d} the maximum flux density in the teeth, f the frequency, and m_{d} the mass of the stator teeth. The iron losses in the stator yoke are

$$P_{\mathrm{Fey}} = 1.5 \cdot p_{15} \left(B_y / 1.5\mathrm{T}\right)^2 \left(f / 50\mathrm{Hz}\right)^{1.5} m_y\,, \tag{4}$$

where B_{y} is the maximum flux density in the yoke, and m_{y} the mass of the yoke. The losses in the permanent magnets are assumed to be 1% of the rated power, i.e., 30 kW. The additional losses are assumed to be 3% of the rated power, i.e., 90 kW. The friction and ventilation losses are

$$P_{\rho} = 10 \cdot D_r \left(l + 0.6 \cdot \tau_p\right)\left(\pi n D_{\mathrm{r}}\right)^2 \left[\mathrm{W}\right], \tag{5}$$

where D_{r} is the outer rotor diameter, τ_p the pole pitch, and n the rotational speed of the rotor. Table 3 gives the nine design parameters to be optimized and their valid ranges.

Table 3. Wind turbine electrical generator design parameters with ranges.

Parameters	Symbols	Ranges
Stator core length including ventilation ducts	l	0.3 – 3.0 m
Stator yoke height	h_{vs}	0.01 – 0.5 m
Stator outer diameter	D_{se}	3.0 – 8.0 m
Stator slot height	h_{slot}	0.07 – 0.3 m
Maximum flux density in air gap	B_{max}	0.4 – 0.9 T
Number of effective conductors in stator slot	z_s	8 – 26
Rotor yoke height	h_{vr}	0.01 – 0.5 m
Number of poles pairs	p	20 – 80
Stator slot width	b_4	0.007 – 0.04 m

Like most of the practical design problems, the design variables of this wind turbine electrical generator are also subject to constraints. A total of five given constraints are provided in Table 4. It can be observed that among them, there are one constraint on the stator tooth width, three constraints on the flux density of stator and rotor yoke and stator tooth, and one constraint on the output power of the wind turbine electrical generator. Therefore, these constraints must be satisfied by the optimized design variables.

Table 4. Wind turbine electrical generator optimization constraints.

Constraint	Value
Stator tooth width	> 8 mm
Stator yoke flux density	< 2.2 T
Rotor yoke flux density	< 2.2 T
Stator tooth flux density	< 2.2 T
Maximum output power	> 4.8 MW

A hybrid HS method: HS-PBIL

A hybrid HS approach, HS-PBIL, is recently proposed by the authors of this paper on the basis of merging the HS together with the Population-Based Incremental Learning (PBIL) [8]. As aforementioned, the new HM members in the HS method can be obtained in two essential ways: combination/mutation of the existing HM members and random generation. In our HS-PBIL, the PBIL-based candidate generation strategy is utilized. That is, the PBIL is used here to generate random but more efficient HM member candidates, as illustrated in Figure 5.

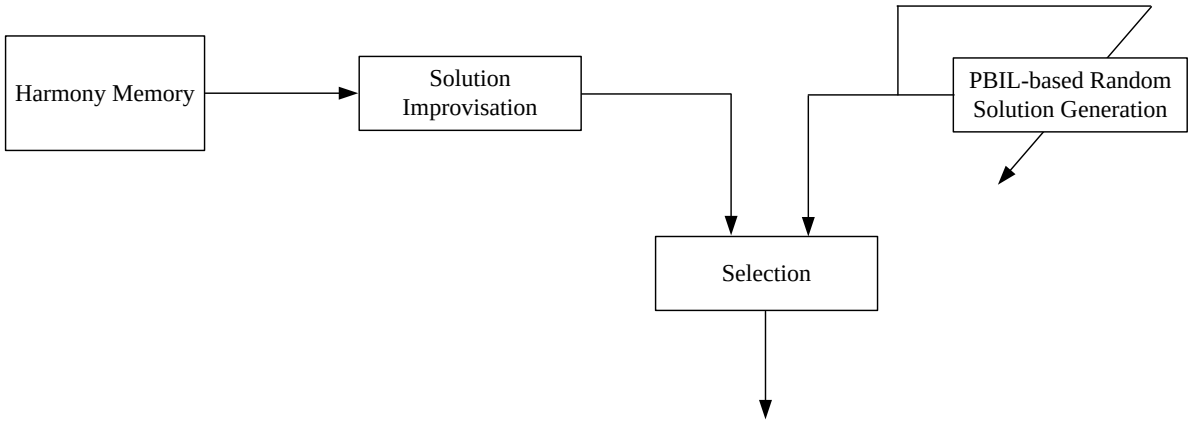

Figure 5. New solution candidates' generation in HS-PBIL.

More precisely, in the HS-PBIL, when a random solution candidate is needed according to the HMCR, it is not generated in a 'pure' random way like in the regular HS method. Instead, it is controlled by a Gaussian distribution function $N(c,\sigma)$ of the PBIL with adaptive parameters of center c and variance σ. Note that both the two parameters can be adjusted on the basis of the PBIL strategy. A dynamical pool containing these solution candidates is also constructed in our HS-PBIL. Let L denote the size of this pool. Every solution candidate created from $N(c,\sigma)$ is not only used as a new individual for the evolution of the HS, but also is stored in the pool. When L solutions have been obtained, the K best ones are selected, and applied to fine-tune c and σ. After the update of $N(c,\sigma)$, all the individuals are expunged from the pool, and fresh solution candidates are continuously generated based on the updated c and σ, and are then used to build up a new pool. We emphasize that this iterative procedure can be implemented in parallel with the HS method. Apparently, the convergence of c and σ leads to the gradual generation of improved solution candidates. Compared with the ones from the random generation strategy, they are supposed to be of higher quality. Thus, with the guidance of the PBIL, the solution candidates created in such a way can gain better and better fitness, which, in return, enhance the overall convergence property of the original HS method. Additionally, the employment of the PBIL only adds a moderate computational burden to the HS, due to its simplicity in nature.

HS-PBIL-based optimal wind turbine electrical generator design

In our simulations, both the original HS and HS-PBIL are applied to deal with the aforementioned wind turbine electrical generator design problem. A penalty function approach is used here to handle the existing constraints in Table 4. A rounding function is utilized on the real-valued solution candidates in the HS and HS-PBIL so that the integer design variables can be obtained. A performance comparison between these two HS methods is made here. After a total of 1,000 independent trials have been run, the average convergence procedures of the HS and HS-PBIL within 1,000 and 10,000 Number of Function Evaluations (NFE) are illustrated in Figures 6 and 7, respectively. Figures 8 and 9 show respectively the corresponding optimal costs acquired by them

during the 1,000 trials. Note that these costs have been ranked. Tables 5 and 6 present the optimal wind turbine electrical generator parameters and corresponding costs obtained by the HS and HS-PBIL after 1,000 and 10,000 NFE, respectively. The best costs, worst costs, and average costs in these two cases are also summarized in Tables 7 and 8. Obviously, compared with the original HS, the HS-PBIL can converge in a faster way, and achieve moderately improved average optimization results within the same NFE, due to the PBIL-based efficient generation of new HM member candidates. As a matter of fact, the best, worst, and average costs acquired by the HS-PBIL are all better than that by the regular HS method. Particularly, the HS-PBIL offers the performance improvements of about 1.6% and 0.1% in the average optimized cost, as shown in Tables 7 and 8, respectively.

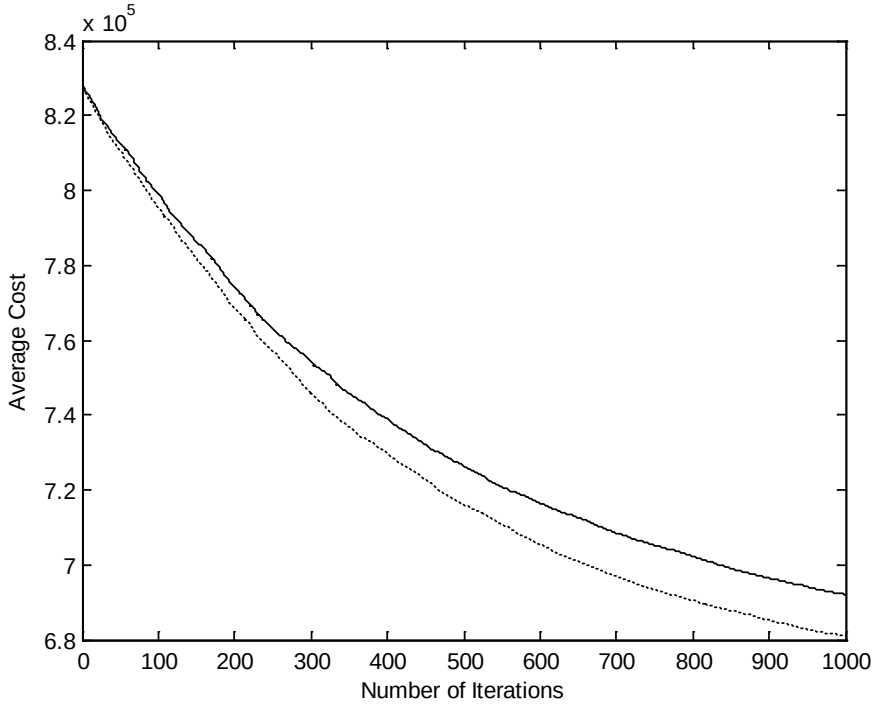

Figure 6. Average convergence procedures of HS and HS-PBIL within 1,000 NFE (solid line: HS, dotted line: HS-PBIL).

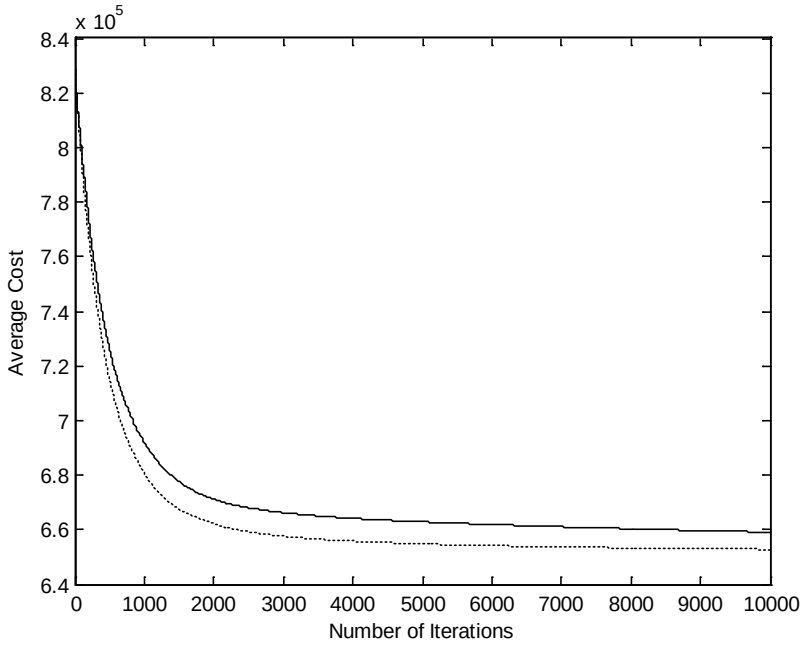

Figure 7. Average convergence procedures of HS and HS-PBIL within 10,000 NFE (solid line: HS, dotted line: HS-PBIL).

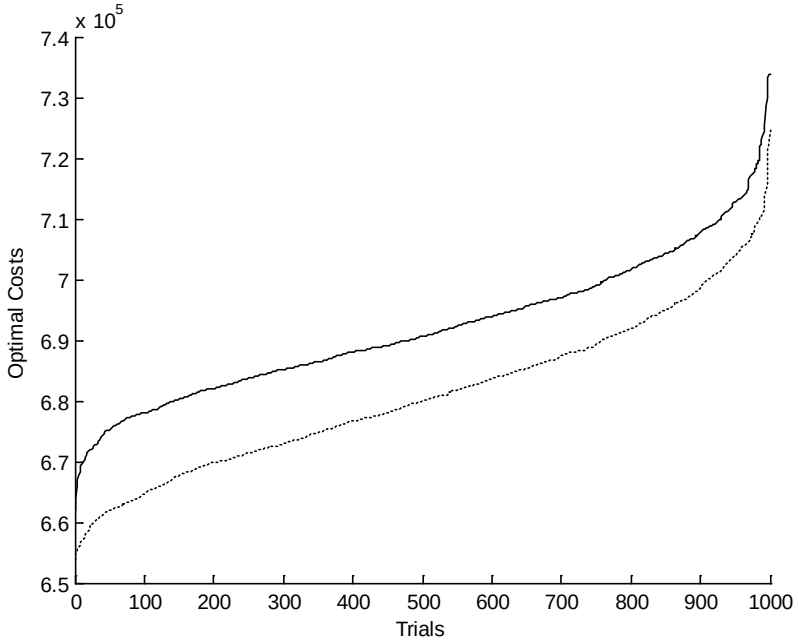

Figure 8. Optimal costs acquired by HS and HS-PBIL within 1,000 NFE (solid line: HS, dotted line: HS-PBIL).

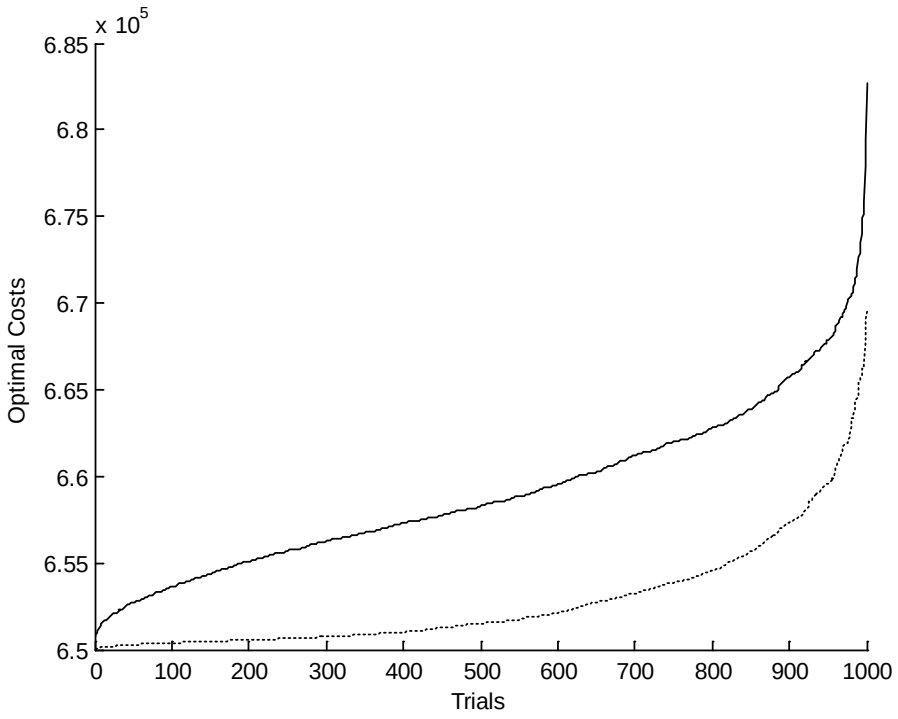

Figure 9. Optimal costs acquired by HS and HS-PBIL within 10,000 NFE (solid line: HS, dotted line: HS-PBIL).

Table 5. Optimal parameters and costs acquired by HS and HS-PBIL within 1,000 NFE.

Parameter	HS	HS-PBIL
l	0.5345	0.5206
h_{vs}	0.0536	0.0470
D_{se}	7.6655	7.7067
h_{slot}	0.1180	0.1044
$B_{\max}$	0.7258	0.7069
z_s	20	20
h_{vr}	0.0401	0.0278
p	66	74
b_4	0.0118	0.0125
Costs	6.6199×10^5	6.5374×10^5

Table 6. Optimal parameters and costs acquired by HS and HS-PBIL within 10,000 NFE.

Parameter	HS	HS-PBIL
l	0.5620	0.4852
h_{vs}	0.0371	0.0374
D_{se}	7.7434	7.9936
h_{slot}	0.0969	0.1026
$B_{\max}$	0.6775	0.6841
z_s	20	22
h_{vr}	0.0263	0.0262
p	72	75
b_4	0.0133	0.0134
Costs	6.5076×10^5	6.5007×10^5

Table 7. Best, worst, and average costs acquired by HS and HS-PBIL within 1,000 NFE.

Cost	HS	HS-PBIL
Best Costs	6.6199×10^5	6.5374×10^5
Worst Costs	7.3380×10^5	7.2472×10^5
Average Costs	6.9199×10^5	6.8100×10^5

Table 8. Best, worst, and average costs acquired by HS and HS-PBIL within 10,000 NFE.

Cost	HS	HS-PBIL
Best Costs	6.5076×10^5	6.5007×10^5
Worst Costs	6.8268×10^5	6.6957×10^5
Average Costs	6.5912×10^5	6.5282×10^5

To further examine and compare the optimization capabilities of the HS and HS-PBIL, four optimization goals, i.e., 6.9×10^5, 6.8×10^5, 6.7×10^5, and 6.6×10^5 are set beforehand. The iteration procedures of the HS and HS-PBIL are terminated, and the NFE used are recorded, after these targeted goals have been achieved. Especially, they can be considered to fail, if the goals are still not reached after 10,000 NFE. Again, 1,000 separate trials have been made in our simulations. The average NFE of the HS and HS-PBIL for achieving the above four optimization goals are provided in Table 9. On the other hand, Table 10 gives the average numbers of failures of the two methods. We can observe from both Tables 9 and 10 that the average NFE used by the HS-PBIL for achieving the same goals are only about 80% of that of the HS. Additionally, the average numbers of failures of the former are much smaller than that of the latter. To summarize, it is obvious that the proposed HS-PBIL is well capable of outperforming the original HS method in coping with this challenging engineering optimization problem. We emphasize that with the involvement of the PBIL, the computational complexity of our

hybrid HS-PBIL is naturally higher than that of the regular HS method. Therefore, the former should cost more computation time than the latter. However, as we know that the calculation of the function evaluation is actually the major issue in the total computation time of a large number of evolutionary computation methods, which is always the most time consuming part. Since the HS-PBIL can save a remarkable numbers of the function evaluations, the overall 'wall clock time' consumed by the HS-PBIL-based optimization algorithm is supposed to be much shorter than that by the HS-based one.

Table 9. Average NFE of HS and HS-PBIL for achieving optimization goals.

Optimization goal	6.9×10^5	6.8×10^5	6.7×10^5	6.6×10^5
HS	1,045	1,474	2,578	5,141
HS-PBIL	832	1,051	1,519	2,552

Table 10. Average numbers of failures of HS and HS-PBIL in achieving optimization goals.

Optimization goal	6.9×10^5	6.8×10^5	6.7×10^5	6.6×10^5
HS	0	1	26	375
HS-PBIL	0	0	2	57

Conclusions

In our paper, the fundamentals of the HS method are first introduced. Next, both the basic HS algorithm and some typical variants are discussed in details. Finally, as an illustrative example here, a modified HS proposed by the authors, HS-PBIL, is employed for handling a real-world optimal wind turbine electrical generator design problem. This engineering example has successfully verified and demonstrated the optimization effectiveness of the HS method.

Acknowledgments

This research work was funded by the Academy of Finland under Grants 135225, 127299, and 137837 and Finnish Funding Agency for Technology and Innovation (TEKES). The authors would like to thank the editors and anonymous reviewers for their insightful comments and constructive suggestions that have improved the paper.

References

[1] H. Ceylan and H. Ceylan, "A hybrid harmony search and TRANSYT hill climbing algorithm for signalized stochastic equilibrium transportation networks," *Transportation Research Part C: Emerging Technologies*, vol. 25, pp. 152–167, 2012. DOI: 10.1016/j.trc.2012.05.007

[2] Y. M. Cheng, L. Li, T. Lansivaara, S. C. Chi, and Y. J. Sun, "Minimization of factor of safety using different slip surface generation methods and an improved

harmony search minimization algorithm," *Engineering Optimization*, vol. 40, no. 2, pp. 95–115, 2008.

[3] S. Das, A. Mukhopadhyay, A. Roy, A. Abraham, and B. K. Panigrahi, "Exploratory power of the harmony search algorithm: analysis and improvements for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, no. 1, pp. 89–106, 2011. DOI: 10.1109/TSMCB.2010.2046035

[4] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*. West Sussex, England: John Wiley & Sons Ltd, 2005.

[5] X. Z. Gao, X. Wang, and S. J. Ovaska, "Uni-modal and multi-modal optimization using modified harmony search methods," International Journal of Innovative Computing, Information and Control, vol. 5, no. 10A, pp. 2985–2996, 2009.

[6] X. Z. Gao, X. Wang, S. J. Ovaska, and H. Xu, "A modified harmony search method in constrained optimization," International Journal of Innovative Computing, Information and Control, vol. 6, no. 9, pp. 4235–4247, 2010.

[7] X. Z. Gao, X. Wang, T. Jokinen, S. J. Ovaska, A. Arkkio, and K. Zenger, "A hybrid optimization method for wind generator design," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 6, pp. 4347–4373, 2012.

[8] X. Z. Gao, X. Wang, S. J. Ovaska, and K. Zenger, "A hybrid optimization method of harmony search and opposition-based learning," *Engineering Optimization*, vol. 44, no. 8, pp. 895–914, 2012. DOI: 10.1080/0305215X.2011.628387

[9] X. Z. Gao, X. Wang, T. Jokinen, S. J. Ovaska, A. Arkkio, and K. Zenger, "A hybrid PBIL-based harmony search method," *Neural Computing and Applications*, vol. 21, no. 5, pp. 1071–1083, 2012. DOI: 10.1007/s00521-011-0675-6

[10] 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.

[11] 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.

[12] Z. W. Geem, "Novel derivative of harmony search algorithm for discrete design variables," *Applied Mathematics and Computation*, vol. 199, no. 1, pp. 223–230, 2008. DOI: 10.1016/j.amc.2007.09.049

[13] Z. W. Geem, "Particle-swarm harmony search for water network design," *Engineering Optimization*, vol. 41, no. 4, pp. 297–311, 2009. DOI: 10.1080/03052150802449227

[14] Z. W. Geem and K.-B. Sim, "Parameter-setting-free harmony search algorithm," *Applied Mathematics and Computation*, vol. 217, no. 8, pp. 3881–3889, 2010. DOI: 10.1016/j.amc.2010.09.049

[15] Z. W. Geem (ed.), *Music-inspired Harmony Search Algorithm*, Springer-Verlag, Berlin, Germany, 2009.

[16] C. Grosan, A. Abraham, and H. Ishibuchi (Eds.), *Hybrid Evolutionary Algorithms*, Berlin, Germany: Springer-Verlag, 2007.

[17] O. Hasancebi, F. Erdal, and M. P. Saka, "Adaptive harmony search method for structural optimization," *ASCE Journal of Structural Engineering*, vol. 136, no. 4, pp. 419–431, 2010. DOI: 10.1061/(ASCE)ST.1943-541X.0000128

[18] K. S. Lee 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. DOI: 10.1016/j.compstruc.2004.01.002

[19] 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. 3902–3933, 2005. DOI: 10.1016/j.cma.2004.09.007

[20] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, Perth, Australia, December 1995, pp. 1942–1945.

[21] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1567–1579, 2007. DOI: 10.1016/j.amc.2006.11.033

[22] M. G. H. Omran and M. Mahdavi, "Global-best harmony search," *Applied Mathematics and Computation*, vol. 198, no. 2, pp. 643–656, 2008. DOI: 10.1016/j.amc.2007.09.004

[23] Q.-K. Pan, P. N. Suganthan, J. J. Liang, and M. F. Tasgetiren, "A local-best harmony search algorithm with dynamic subpopulations," *Engineering Optimization*, vol. 42, no. 2, pp. 101–117, 2010. DOI: 10.1080/03052150903104366

[24] R. Poli and W. B. Langdon, *Foundations of Genetic Programming*. Berlin, Germany: Springer-Verlag, 2002.

[25] J. Pyrhönen, T. Jokinen, and V. Hrabovcová, *Design of Rotating Electrical Machines*. West Sussex, UK: John Wiley & Sons Ltd, 2008.

[26] X. Wang, X. Z. Gao, and S. J. Ovaska, "Fusion of clonal selection algorithm and harmony search method in optimization of fuzzy classification systems," *International Journal of Bio-Inspired Computation*, vol. 1, no. 1–2, pp. 80–88, 2009. DOI: 10.1504/IJBIC.2009.022776

[27] X. Wang, X. Z. Gao, and K. Zenger, *An Introduction to Harmony Search Optimization Method*, Springer-Verlag, Springer Briefs in Computational Intelligence, 2014.

[28] C.-M. Wang and Y.-F. Huang, “Self-adaptive harmony search algorithm for optimization,” *Expert Systems with Applications*, vol. 37, no. 4, pp. 2826–2837, 2010. DOI: 10.1016/j.eswa.2009.09.008

Xiao-Zhi Gao
Machine Vision and Pattern Recognition Laboratory
Lappeenranta University of Technology, Lappeenranta, Finland
E-mail: xiao.z.gao@gmail.com

Xiaolei Wang
Dynavio Oy, Espoo, Finland
E-mail: wxlbless@yahoo.com

Kai Zenger
Department of Electrical Engineering and Automation
Aalto University School of Electrical Engineering, Espoo, Finland
E-mail: kai.zenger@aalto.fi