International Journal of Innovative Computing, Information and Control
Volume **8**, Number **6**, June **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 4347–4373

A HYBRID OPTIMIZATION METHOD FOR WIND GENERATOR DESIGN

Xiao-Zhi Gao[1], Xiaolei Wang[1], Tapani Jokinen[2], Seppo Jari Ovaska[2]
Antero Arkkio[2] and Kai Zenger[1]

[1]Department of Automation and Systems Technology
School of Electrical Engineering
Aalto University
P. O. Box 15500, FI-00076 Aalto, Finland
{ xiao-zhi.gao; xiaolei.wang; kai.zenger }@aalto.fi

[2]Department of Electrical Engineering
School of Electrical Engineering
Aalto University
P. O. Box 13000, FI-00076 Aalto, Finland
{ tapani.jokinen; seppo.ovaska; antero.arkkio }@aalto.fi

Received March 2011; revised July 2011

Abstract. *The Harmony Search (HS) method is an emerging meta-heuristic optimization algorithm. However, like most of the evolutionary computation techniques, the HS does not store or utilize the useful knowledge gained during the search procedure in an efficient way. In this paper, we propose and study a hybrid optimization approach, in which the HS is merged together with the Cultural Algorithm (CA). Our modified HS method, namely HS-CA, has the interesting feature of embedded problem-solving knowledge. Simulations of some typical benchmark problems demonstrate that the HS-CA can yield a superior optimization performance over the regular HS algorithm. We also apply the proposed HS-CA in a case study of the optimal wind generator design to further examine its effectiveness.*

Keywords: Harmony search (HS), Cultural algorithm (CA), Search knowledge, Hybrid optimization methods, Wind generator optimization

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], pipe network optimization [4], Magnetic Resonance Imaging (MRI) brain segmentation [5], and redundancy optimization problems of electrical and mechanical systems [6]. A lot of modified HS algorithms have been studied in the past decade. For example, based on the continuous HS, Geem proposes a discrete version by introducing the stochastic derivatives for the discrete variables involved [7]. Omran and Mahdavi embed the ideas borrowed from swarm intelligence into the regular HS, and develop an improved HS technique: Global-best HS [8]. Several new variants of the HS method are also introduced by the authors of the present paper [9-11].

Theoretical research on the working principles and search mechanism of the HS method has 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 [12]. Based on their analysis work, they further propose a modified HS algorithm, Exploratory HS (EHS), in which the bandwidth (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 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, empirical study has shown that the original HS method sometimes suffers from a slow search speed [2]. One main reason behind this weakness is that the problem-dependent knowledge that can be collected from the evolving process is not considered in the HS method. It has been proved that the employment of an effective mechanism to represent, acquire, store, and utilize the search knowledge can bias and accelerate the convergence of various evolutionary computation methods [13]. As a matter of fact, Reynolds proposes a distinguishing Cultural Algorithm (CA), in which there exist two spaces, population space and belief space, for the acquisition and deployment of search knowledge [14]. The problem-solving experiences from the individuals in the population space are first extracted and stored in the belief space, and are next used to influence the evolution of the population space. Therefore, in this paper, we develop and explore a hybridization of the HS and CA: HS-CA. In our HS-CA, the situational and normative knowledge from the CA is applied to properly adjust and guide the mutation of the new solution candidates in the HS method so that an improved convergence performance can be achieved. The application of the proposed HS-CA in the optimal design of wind generator is also investigated.

The rest of this paper is organized as follows. We briefly introduce the background of both the HS and CA in Sections 2 and 3, respectively. In Section 4, by embedding the CA into the HS method, we propose a new hybrid optimization algorithm: HS-CA. The principle of the developed HS-CA is discussed in detail here. Simulations of a total of 15 nonlinear functions and four engineering optimization problems are demonstrated in Section 5. The wind generator design problem is presented and explained in Section 6, and Section 7 demonstrates the investigation results of the application of the HS-CA in this case study. Finally, in Section 8, we conclude our paper with some remarks and conclusions.

2. Harmony Search (HS) 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 explicit 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 HMS 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 $[x_1^i, x_2^i, \cdots, x_n^i]$ $(i = 1, 2, \cdots, HMS)$ is a solution candidate. HMS is typically set to be between 10 and 100.

Step 2. Improvise a new solution $[x_1', x_2', \cdots, x_n']$ 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 jth 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) [15,16] 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, Particle Swarm Optimization (PSO) [17-19], and Differential Evolution (DE) [20,21], the HS method is a random search technique. It does not require any prior domain information, such as the gradient 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 characteristics of algorithm simplicity.

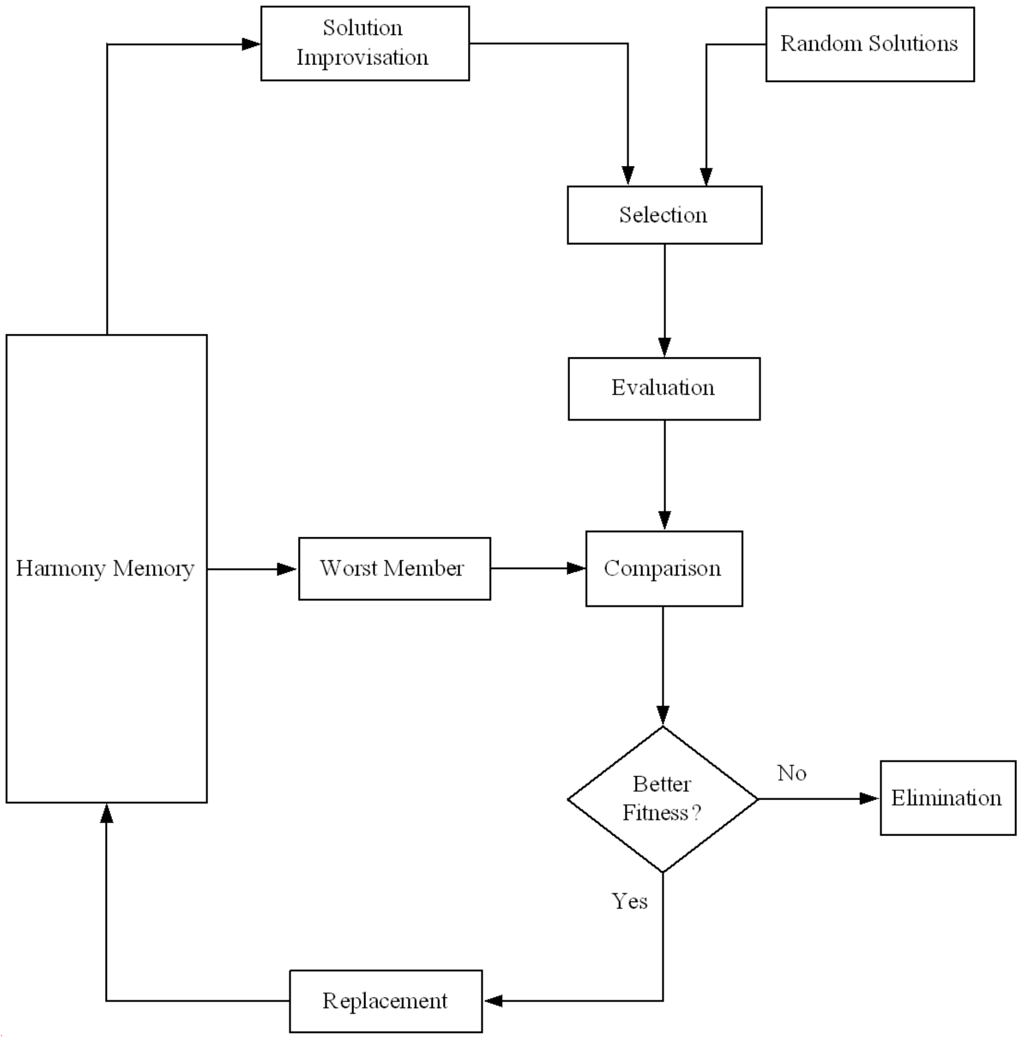


FIGURE 1. Harmony search (HS) method

3. **Cultural Algorithm (CA).** As proposed by Reynolds, the CA is a dual inheritance system with its evolution on two different levels: population level and belief level [22]. The culture in the CA refers to the beliefs or experiences, which can be gained from and used to direct the evolving individuals. The basic framework of the CA is shown in Figure 2, where there are two spaces, population space and belief space, interacting with each other. Similar to the regular evolutionary computation algorithms, the evolution of the individuals in the population space is based on such operations as mutation and crossover [23,24]. However, besides the population space, the CA has the belief space that can acquire and store certain domain knowledge from the individuals in the population space. More precisely, the most fitted individuals in the population space are selected by the 'Accept' function to update the knowledge in the belief space. The function of 'Influence' takes advantage of this kind of knowledge to guide the evolution of the population space. The knowledge in the belief space is classified into two essential types: situational knowledge and normative knowledge. The situational knowledge is actually the best exemplar chosen from the population space, while the normative knowledge contains the useful information concerning the search regions, where the above-average individuals may exist. In a word, both the situational knowledge and normative knowledge in the belief space can control and improve the population space.

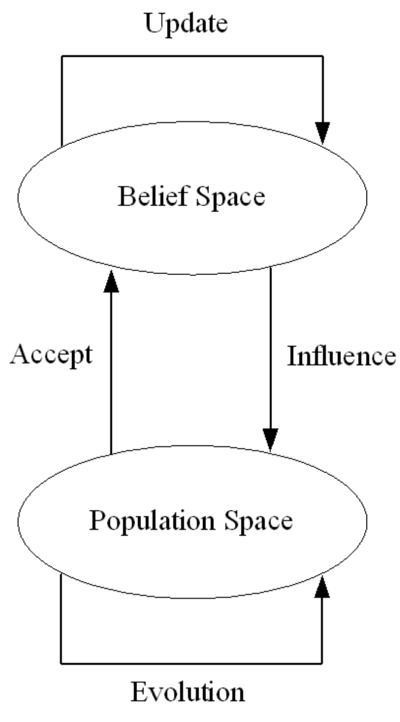


Figure 2. Cultural algorithm (CA)

The pseudo codes of the aforementioned CA can be presented as follows [13]:

Step 1: Set $t = 0$.
Step 2: Initialize population space P^t.
Step 3: Initialize belief space B^t.
Step 4: Update belief space B^t using $Accept(P^t)$.
Step 5: Update population space P^t using $Influence(B^t)$.
Step 6: $t = t + 1$.
Step 7: Return back to Step 4 until a given termination criterion is met.

It is concluded from the above descriptions that the unique belief space plays a key role in the CA. In the next section, we will integrate the knowledge management and utilization mechanisms of the CA into the original HS method, and propose a hybrid optimization approach: HS-CA. This HS-CA uses the belief space to enhance the generation of new HM members so that their fitness can be significantly improved.

4. **Fusion of HS and CA: HS-CA.** From Section 2, we can observe that based on the PAR, the new solution candidates improvised from the existing HM members may go through a mutation procedure. The mutation is generally a 'blind' and local exploration in the search space. Unfortunately, it is well known that the pure random-based mutation is not effective in guiding the evolution of individuals. Thus, the CA can be employed to influence the direction and size of the mutation in the HS method. That is to say, the knowledge in the belief space of the CA is first extracted from the HM, and next used to direct the mutation of the new solutions. Inspired by this idea, we here propose a hybridization of the HS method and CA: HS-CA, as illustrated in Figure 3.

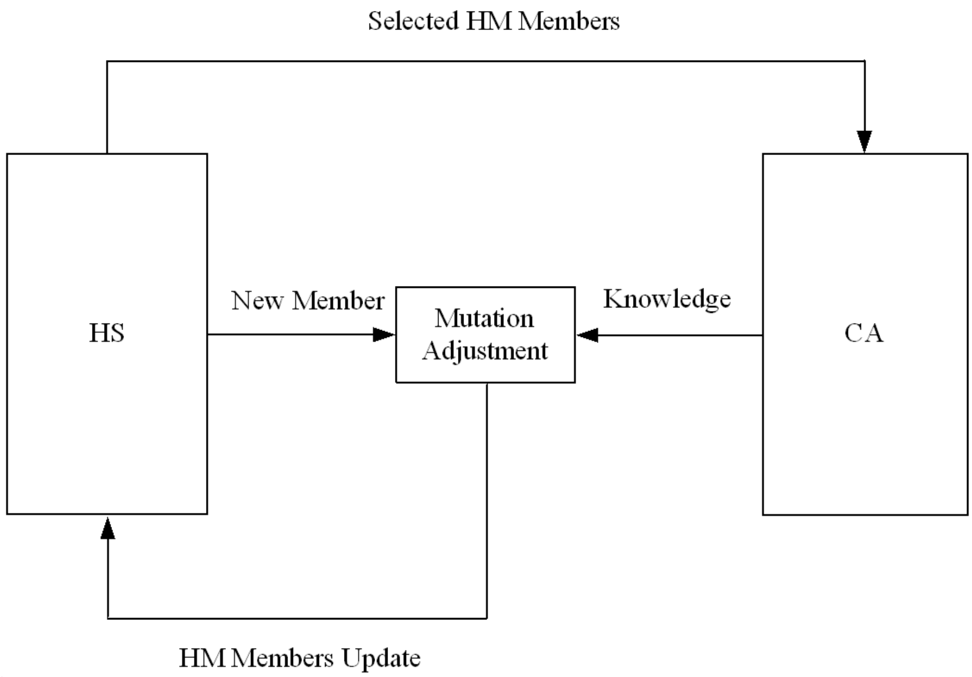


FIGURE 3. Hybridization of HS and CA: HS-CA

To put it into more details, suppose any individual in the HS-CA is represented as $\mathbf{x}_i^t = \left[x_{i,1}^t, x_{i,2}^t, \cdots, x_{i,N}^t\right]$, where N is the dimension of the solution space, and t is the iteration step. All the individuals are evaluated by fitness function $f(\cdot)$. The optimization goal is, therefore, to find the optimal $\mathbf{x}^t$ that can maximize $f(\cdot)$. The situational knowledge of the CA is denoted as $\mathbf{S}^t$. As discussed above, $\mathbf{S}^t$ should be the best individual selected from the HM, and is updated as follows:

$$\begin{array}{ll} \mathbf{S}^{t+1} = \mathbf{x}_{best}^t, & \text{if } f(\mathbf{x}_{best}^t) > f(\mathbf{S}^t), \\ \mathbf{S}^{t+1} = \mathbf{S}^t, & \text{otherwise.} \end{array}$$

The normative knowledge of the CA is represented by N intervals for the dimensions of the individuals. Each interval is associated with a triple $\langle I_i^t, L_i^t, U_i^t \rangle$. I_i^t is a closed interval with lower bound l_i^t and upper bound u_i^t. L_i^t and U_i^t are the 'scores' for l_i^t and u_i^t, respectively. Usually, l_i^t and u_i^t are initialized by the given problem domain, and both L_i^t and U_i^t are initially set to be $+\infty$. $\langle I_i^t, L_i^t, U_i^t \rangle$ can be updated by the top HM members

$\mathbf{x}_1^t, \mathbf{x}_2^t, \cdots, \mathbf{x}_M^t$, where M is the number of the selected candidates, e.g., $M = HMS \times Top$, where Top is a percentage constant. For $\mathbf{x}_j^t$, we have:

$$\begin{array}{ll} l_i^{t+1} = x_{j,i}^t, & \text{if } x_{j,i}^t \leq l_i^t \text{ or } f(\mathbf{x}_j^t) > L_i^t, \\ l_i^{t+1} = l_i^t, & \text{otherwise.} \\ L_i^{t+1} = f(\mathbf{x}_j^t), & \text{if } x_{j,i}^t \leq l_i^t \text{ or } f(\mathbf{x}_j^t) > L_i^t, \\ L_i^{t+1} = L_i^t, & \text{otherwise.} \\ u_i^{t+1} = x_{j,i}^t, & \text{if } x_{j,i}^t \geq u_i^t \text{ or } f(\mathbf{x}_j^t) > U_i^t, \\ u_i^{t+1} = u_i^t, & \text{otherwise.} \\ U_i^{t+1} = f(\mathbf{x}_j^t), & \text{if } x_{j,i}^t \geq u_i^t \text{ or } f(\mathbf{x}_j^t) > U_i^t, \\ U_i^{t+1} = U_i^t, & \text{otherwise.} \end{array}$$

Note that the update policy of the normative knowledge is that it should be conservative and progressive to narrow and widen respectively the interval I_i^t. Both the situational knowledge and normative knowledge is utilized to determine the mutation operation in the HS of our HS-CA.

Suppose $\mathbf{x}_*$ is a new solution improvised by the random combination of the components of the HM members, and $\mathbf{x}'_*$ is the mutated version of $\mathbf{x}_*$. If a conventional mutation operator is used, $\mathbf{x}'_*$ can be generated as:

$$\mathbf{x}'_* = \mathbf{x}_* + Rand, \tag{2}$$

where $Rand$ represents a random number generator. However, compared with (2), the $Influence$ function of the CA on the mutation is more goal-directed. There are four basic types of $Influence$ functions, N_s, S_d, $N_s + S_d$, and $N_s + N_d$, depending on the ways how the situational knowledge and normative knowledge is employed to guide the mutation. They are explained in details as the following.

For N_s,

$$\mathbf{x}'_{*_i} = \mathbf{x}_{*_i} + Size(I_i) \times N(0,1), \tag{3}$$

where $Size(I_i)$ is the size of the belief interval I_i for the ith dimension of $\mathbf{x}_*$, and $N(0,1)$ is a random number with the normal distribution, whose mean and standard derivation are 0 and 1, respectively.

For S_d,

$$\begin{array}{ll} \mathbf{x}'_{*_i} = \mathbf{x}_{*_i} + |\sigma_i \times N(0,1)|, & \text{if } \mathbf{x}_{*_i} < \mathbf{S}_i, \\ \mathbf{x}'_{*_i} = \mathbf{x}_{*_i} - |\sigma_i \times N(0,1)|, & \text{if } \mathbf{x}_{*_i} > \mathbf{S}_i, \\ \mathbf{x}'_{*_i} = \mathbf{x}_{*_i} + \sigma_i \times N(0,1), & \text{otherwise,} \end{array}$$

where σ_i is the given mutation step for the ith dimension of $\mathbf{x}_*$, and $\mathbf{S}_i$ is the situational knowledge for the ith dimension of $\mathbf{x}_*$ in the belief space.

For $N_s + S_d$,

$$\begin{array}{ll} \mathbf{x}'_{*_i} = \mathbf{x}_{*_i} + |Size(I_i) \times N(0,1)|, & \text{if } \mathbf{x}_{*_i} < \mathbf{S}_i, \\ \mathbf{x}'_{*_i} = \mathbf{x}_{*_i} - |Size(I_i) \times N(0,1)|, & \text{if } \mathbf{x}_{*_i} > \mathbf{S}_i, \\ \mathbf{x}'_{*_i} = \mathbf{x}_{*_i} + Size(I_i) \times N(0,1), & \text{otherwise.} \end{array}$$

For $N_s + N_d$,

$$\begin{array}{ll} \mathbf{x}'_{*_i} = \mathbf{x}_{*_i} + |Size(I_i) \times N(0,1)|, & \text{if } \mathbf{x}_{*_i} < l_i, \\ \mathbf{x}'_{*_i} = \mathbf{x}_{*_i} - |Size(I_i) \times N(0,1)|, & \text{if } \mathbf{x}_{*_i} > u_i, \\ \mathbf{x}'_{*_i} = \mathbf{x}_{*i} + \beta \times Size(I_i) \times N(0,1), & \text{otherwise,} \end{array}$$

where l_i and u_i are the normative knowledge (lower and upper bounds for the ith dimension of $\mathbf{x}_*$, respectively) in the belief space, and β is a preset small constant.

It should be emphasized that the belief space in the CA can identify the promising regions, where the above-average individuals may be present. The objective of the

Influence function is to promote the newly generated solutions to those regions so that their fitness is improved. Different from the mutation operation in (2), the one of the proposed HS-CA is effectively controlled by the *Influence* function. Therefore, embedding the CA into the HS method can indeed yield a superior optimization performance. In Section 5, we are going to use 15 nonlinear functions and four engineering problems to demonstrate the accelerated convergence of our HS-CA.

5. **Optimization of Nonlinear Functions and Engineering Design.** In this section, we investigate the effectiveness of the proposed HS-CA with some simulation examples of nonlinear functions as well as engineering problems.

5.1. **Nonlinear functions.** The following 15 n-dimensional nonlinear functions, which have been widely used as popular optimization benchmarks [25,26], are employed to compare the optimization (minimization) capabilities between the HS and HS-CA.

Ackley function:

$$f(\mathbf{x}) = -20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)} + 20 - e, \quad x_i \in [-10, 10]. \tag{4}$$

Alpine function:

$$f(x) = \sum_{i=1}^{n} |x_i \sin(x_i) + 0.1x_i|, \quad x \in [-10, 10]. \tag{5}$$

Bohachevsky function:

$$f(x) = \sum_{i=1}^{n} \left[x_i^2 + 2x_{i+1}^2 - 0.3\cos(3\pi x_i) - 0.4\cos(4\pi x_{i+1}) + 0.7\right], \quad x \in [-5.12, 5.12]. \tag{6}$$

De Jong function:

$$f(x) = \sum_{i=1}^{n} ix_i^4, \quad x \in [-2.56, 2.56]. \tag{7}$$

Dixon and Price function:

$$f(\mathbf{x}) = (x_1 + 1)^2 + \sum_{i=1}^{n} i(2x_i^2 - x_{i-1})^2, \quad x_i \in [-10, 10]. \tag{8}$$

Griewank function:

$$f(\mathbf{x}) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\frac{x_i}{\sqrt{i}} + 1, \quad x_i \in [-10, 10]. \tag{9}$$

Hyper Ellipsoid function:

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

Levy function:

$$f(\mathbf{x}) = \sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 \left[1 + \sin^2(3\pi x_{i+1})\right] + (x_n - 1)\left[1 + \sin^2(3\pi x_n)\right], \quad x_i \in [-10, 10]. \tag{11}$$

Michalewicz function:

$$f(\mathbf{x}) = -\sum_{i=1}^{n} \sin(x_i)\left[\sin\left(\frac{ix_i^2}{\pi}\right)\right]^{20}, \quad x_i \in [0, \pi]. \tag{12}$$

Powell function:

$$f(\mathbf{x}) = \sum_{i=1}^{\frac{n}{4}} (x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^2,$$
$$x_i \in [-4, 5]\,. \tag{13}$$

Rastrigin function:

$$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2 + 10 - 10\cos(2\pi x_i), \quad x_i \in [-5.12, 5.12]\,. \tag{14}$$

Rosenbrock function:

$$f(\mathbf{x}) = \sum_{i=1}^{n} 100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2, \quad x_i \in [-10, 10]\,. \tag{15}$$

Sphere function:

$$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2, \quad x_i \in [-10, 10]\,. \tag{16}$$

Trid function:

$$f(\mathbf{x}) = \sum_{i=1}^{n} (x_i - 1)^2 - \sum_{i=2}^{n} x_i x_{i-1}, \quad x_i \in [-100, 100]\,. \tag{17}$$

Zakharov function:

$$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} \frac{ix_i}{2}\right)^2 + \left(\sum_{i=1}^{n} \frac{ix_i}{2}\right)^4, \quad x_i \in [-5, 10]\,. \tag{18}$$

The global minima of all the above functions are at $f(\mathbf{x}) = 0$, except for Michalewicz 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 most optimization systems. 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-CA. Both of them have 100 HM members, i.e., $HMS = 100$, which are always initialized to be equal. The relevant parameters in these two optimization methods are as follows: HMCR = 0.8 and PAR = 0.6. Their evolution procedures are terminated after 10,000 NFE. In the HS-CA, we use the $Influence$ function of $N_s + N_d$ for all the 15 functions, and the corresponding CA parameters of Top and β are given in Table 1.

Tables 2-4 present the optimal solutions acquired by the HS and HS-CA, when $n = 10$, $n = 30$, and $n = 50$, respectively. We stress that the results here are based on the average of 100 independent trials. As an illustrative example, the optimal solutions to the Ackley function ($n = 10$) acquired from the HS and HS-CA are shown in Figure 4(a) and Figure 4(b), respectively. In addition, Figure 5 shows the iteration procedures (average over 100 trials) of these two methods in optimizing the same function. Apparently, compared with the original HS method, for all the 15 test functions, our HS-CA can achieve considerably better optimization results within the same NFE, due to the CA-based utilization of the search knowledge extracted from the HS. That is, the HS-CA has a superior nonlinear function optimization capability.

The above simulation results indeed demonstrate that the HS-CA can outperform the regular HS method. Nevertheless, its performance is unavoidably affected by the CA parameters involved. More specifically, the choice of Top, β, and $Influence$ function play a pivotal role in the optimization capability of the HS-CA. For example, Figure

Table 1. Parameters of Top and β in HS-CA

	Top	β
Ackley Function	10%	0.1
Alpine Function	10%	0.1
Bohachevsky Function	10%	0.1
De Jong Function	10%	0.1
Dixon and Price Function	10%	0.1
Griewank Function	10%	0.1
Hyper Ellipsoid Function	10%	0.1
Levy Function	10%	0.1
Michalewicz Function	10%	0.01
Powell Function	10%	0.1
Rastrigin Function	5%	0.1
Rosenbrock Function	10%	0.1
Sphere Function	10%	0.1
Trid Function	10%	0.2
Zakharov Function	5%	0.1

Table 2. Optimal solutions acquired by HS and HS-CA within 10,000 NFE ($n = 10$)

	HS	HS-CA
Ackley Function	1.4802	1.3516×10^{-12}
Alpine Function	0.0117	2.1185×10^{-4}
Bohachevsky Function	0.7144	0.0105
De Jong Function	8.8887×10^{-6}	2.1577×10^{-45}
Dixon and Price Function	1.4288	0.6613
Griewank Function	21.6018	0.1543
Hyper Ellipsoid Function	137.1776	0.0962
Levy Function	0.0944	6.5504×10^{-24}
Michalewicz Function	-9.0838	-9.5722
Powell Function	0.0378	0.0015
Rastrigin Function	2.8585	0.2498
Rosenbrock Function	44.4376	7.8584
Sphere Function	0.0025	2.7335×10^{-24}
Trid Function	-165.7292	-200.5229
Zakharov Function	2.5722	8.5050×10^{-4}

6 illustrates the iteration procedures of the HS-CA with $Top = 5\%$, $Top = 10\%$, and $Top = 15\%$ for the optimization of the Rastrigin function ($n = 50$). Apparently, different Top values can lead to different convergence characteristics of our HS-CA. We also examine the effect of parameter β on the HS-CA in optimizing the Michalewicz function ($n = 50$), as shown in Figure 7. The HS-CA with $\beta = 0.01$ converges must faster than the ones with $\beta = 0.05$ and $\beta = 0.1$. Unfortunately, there is no analytic way yet for us to choose the best values for these CA parameters, which are often determined based on *trial and error*. The four kinds of the $Influence$ functions for the Rastrigin function optimization are explored in Figure 8. It can be observed that the effectiveness of this HS-CA significantly deteriorates with the $Influence$ functions of N_s and $N_s + S_d$. Reynolds argues that the

Table 3. Optimal solutions acquired by HS and HS-CA within 10,000 NFE ($n = 30$)

	HS	HS-CA
Ackley Function	2.5364	5.165×10^{-5}
Alpine Function	0.5881	0.0015
Bohachevsky Function	4.6559	0.3680
De Jong Function	0.0046	8.2869×10^{-17}
Dixon and Price Function	24.5650	0.8037
Griewank Function	61.9223	0.3622
Hyper Ellipsoid Function	5.6404×10^{8}	1.1386×10^{7}
Levy Function	1.1288	0.0398
Michalewicz Function	-17.2730	-28.6997
Powell Function	3.7738	0.1544
Rastrigin Function	24.0671	11.3442
Rosenbrock Function	251.3008	35.4433
Sphere Function	0.0773	6.0406×10^{-10}
Trid Function	-451.8707	-897.9181
Zakharov Function	261.7912	176.3173

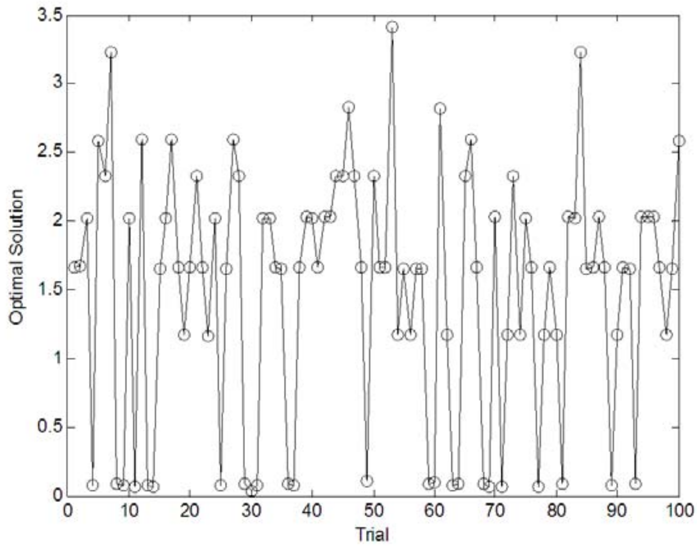
Table 4. Optimal solutions acquired by HS and HS-CA within 10,000 NFE ($n = 50$)

	HS	HS-CA
Ackley Function	3.0282	0.0126
Alpine Function	2.3841	0.0078
Bohachevsky Function	12.4675	1.4351
De Jong Function	10.0609	6.7928×10^{-9}
Dixon and Price Function	415.4205	3.3149
Griewank Function	102.9463	3.2802
Hyper Ellipsoid Function	5.7603×10^{14}	3.2679×10^{11}
Levy Function	3.2569	0.1956
Michalewicz Function	-21.9800	-44.0748
Powell Function	14.2161	1.1277
Rastrigin Function	61.6260	39.7626
Rosenbrock Function	821.9302	91.6834
Sphere Function	0.6496	1.4131×10^{-5}
Trid Function	-185.4805	-996.127
Zakharov Function	581.7118	470.9657

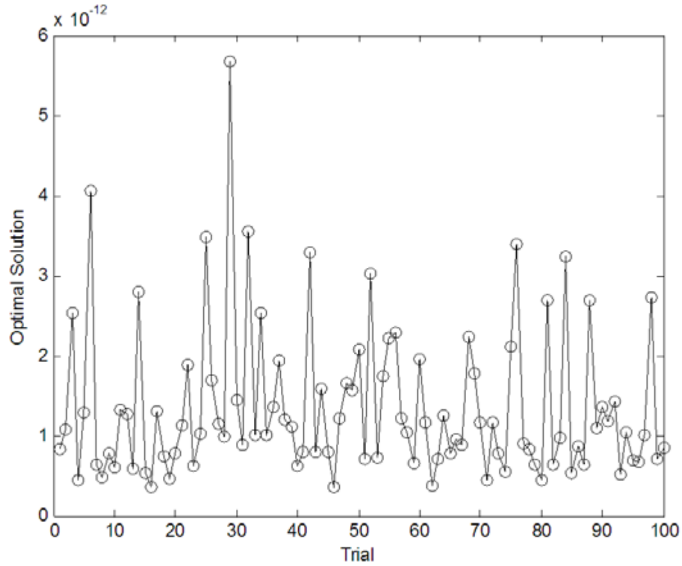
reason behind this may lie in the structures of the functions to be optimized [14]. Further discussions on the $Influence$ function are given in the same paper as well.

5.2. Engineering optimization problems.

Example 5.1. *Minimization of weight of tension/compression spring. This simple but representative engineering optimization problem targets at minimizing the weight of a tension/compression spring, subject to the constraints on its shear stress, surge frequency, and minimum deflection [27]. The parameters to be optimized are the mean coil diameter D, wire diameter d, and number of active coils N. A tension/compression spring with*



(a)



(b)

Figure 4. Optimal solutions to Ackley function ($n = 10$) acquired by HS and HS-CA: (a) HS, (b) HS-CA

these design parameters is shown in Figure 9. The spring design problem can be explained as follows:

$$\textit{minimize } f(\mathbf{x}) = x_1^2 x_2 (x_3 + 2),$$

$$\textit{subject to } g_1(\mathbf{x}) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0,$$

$$g_2(\mathbf{x}) = \frac{4x_2^2 - x_1 x_2}{12566(x_1^3 x_2 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \leq 0,$$

$$g_3(\mathbf{x}) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0,$$

$$g_4(\mathbf{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0,$$

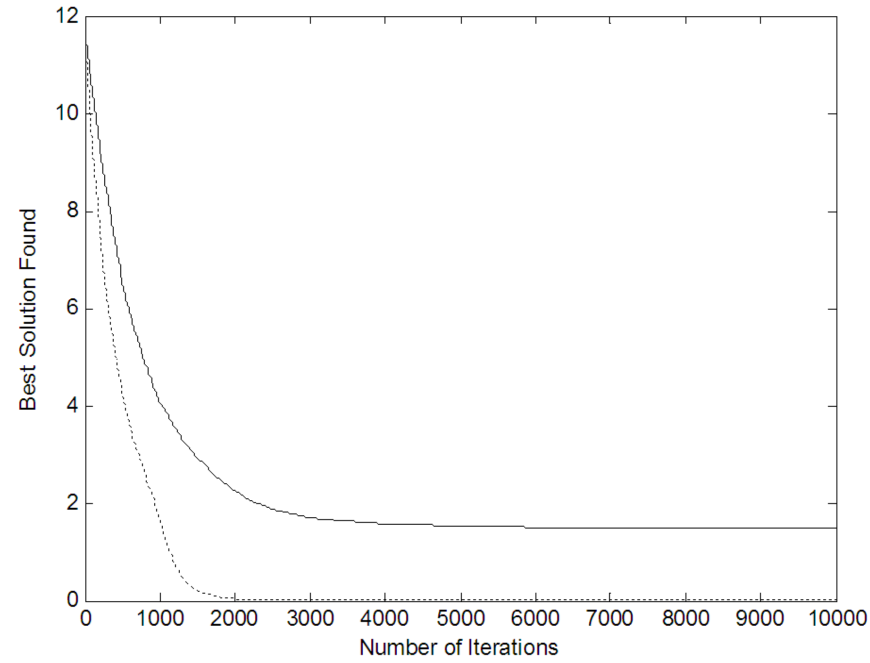


FIGURE 5. Optimization procedures of Ackley function ($n = 10$) with HS (solid line) and HS-CA (dotted line)

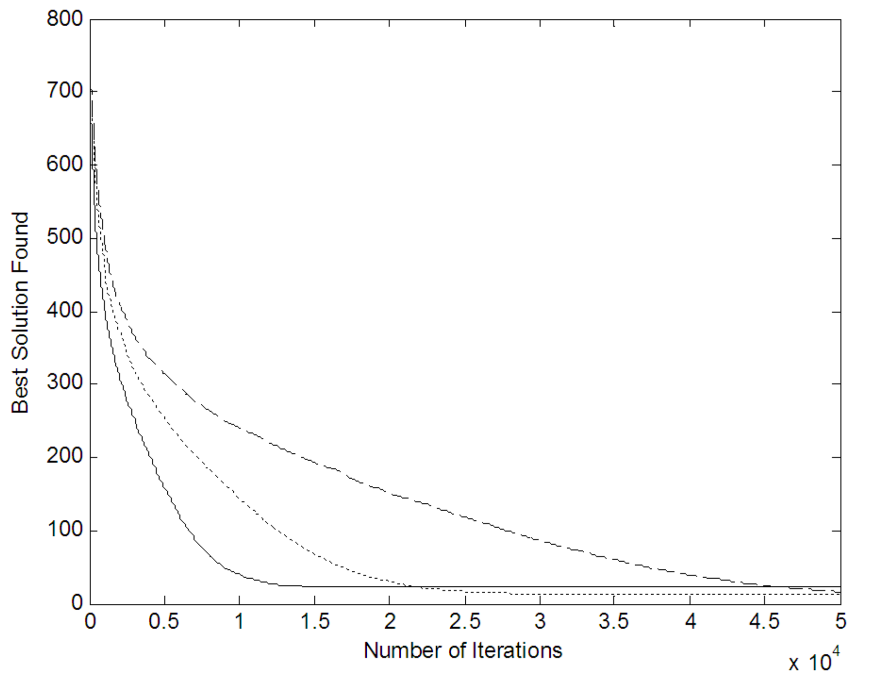


FIGURE 6. Optimization procedures of Rastrigin function ($n = 50$) with HS-CA (solid line: $Top = 5\%$, dotted line: $Top = 10\%$, dashed line: $Top = 15\%$)

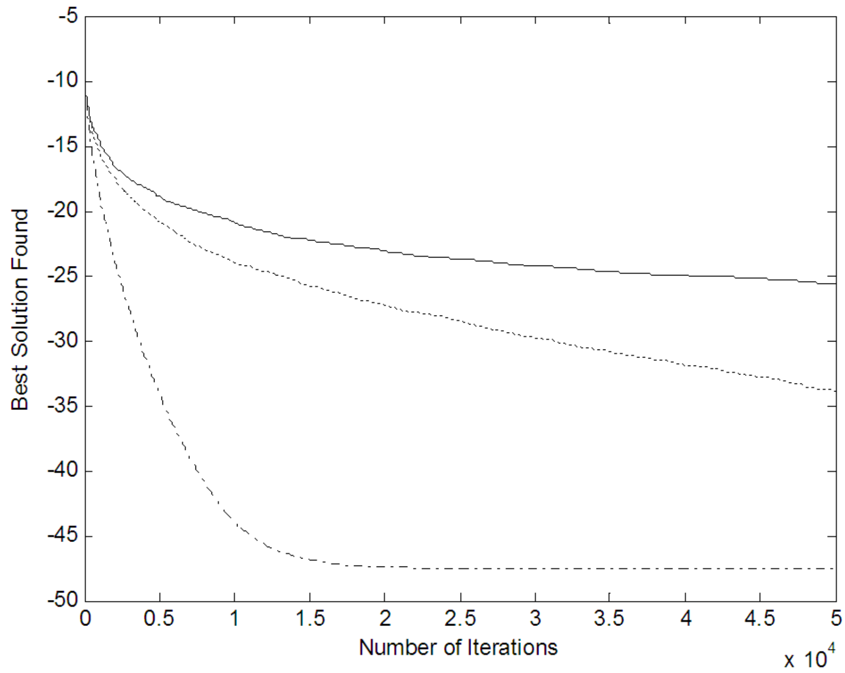


FIGURE 7. Optimization procedures of Michalewicz function ($n = 50$) with HS-CA (solid line: $\beta = 0.1$, dotted line: $\beta = 0.05$, dashed line: $\beta = 0.01$)

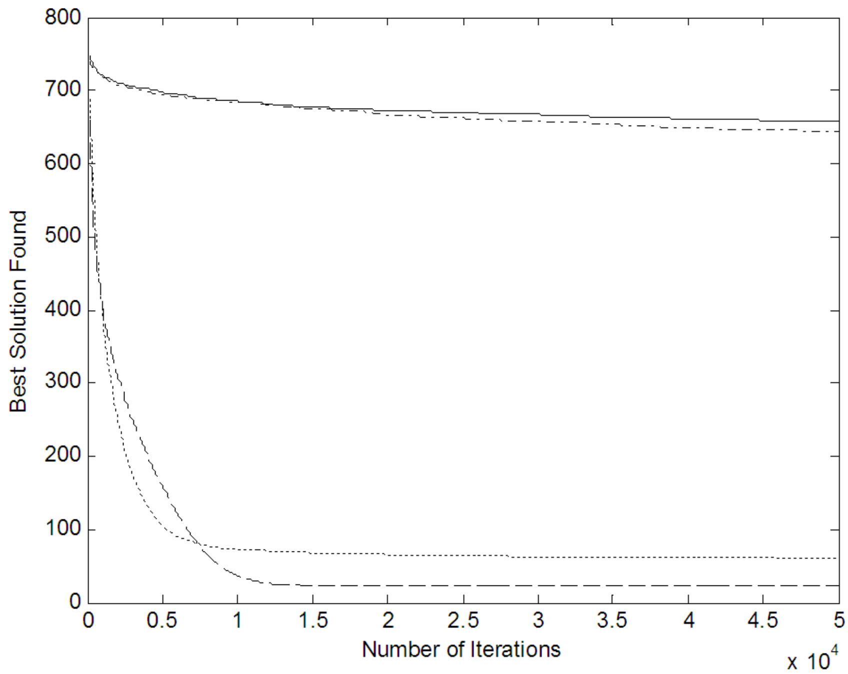


FIGURE 8. Optimization procedures of Rastrigin function ($n = 50$) with HS-CA (solid line: N_s, dotted line: S_d, dashed-dotted line: $N_s + S_d$, dashed line: $N_s + N_d$)

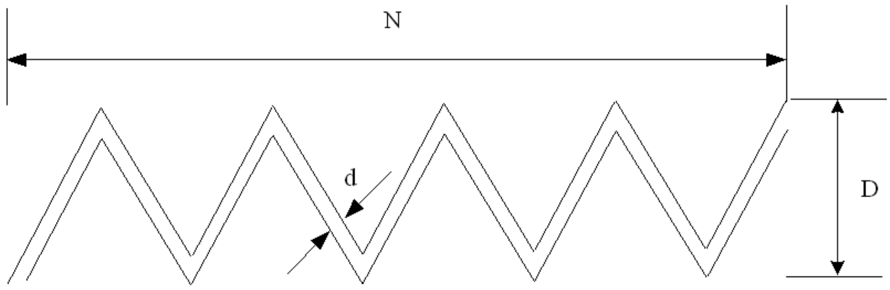


FIGURE 9. Tension/compression spring in Example 5.1

where x_1 *is* D, x_2 *is* d, *and* x_3 *is* N.

To compare the HS and HS-CA in dealing with the above optimal spring design problem, we use a total of 30,000 function evaluations and 100 trials. The Influence function of $N_s + N_d$ *is considered in our HS-CA, and* β *is 0.2. All the other parameters used in both the HS and HS-CA are set to be the same. The optimization results (average over 100 trials) of these two methods are summarized in Table 5. Apparently, the HS-CA can provide us with a better objective function value than the HS:* $f(\mathbf{x}) = 0.012989$ vs. $f(\mathbf{x}) = 0.013201$.

TABLE 5. Optimization results of weight of tension/compression spring with HS and HS-CA

	x_1	x_2	x_3	$f(\mathbf{x})$
HS	0.054645	0.433579	8.743544	0.013201
HS-CA	0.055071	0.445656	7.913870	0.012989

Example 5.2. *Optimal welded beam design. The optimal design of the welded beam has been an important benchmark for modern optimization methods [28]. The goal here is to minimize the fabricating cost of the welded beam, subject to the constraints on the shear stress of the weld,* $\tau(\mathbf{x})$, *bending stress on the beam,* $\sigma(\mathbf{x})$, *buckling load on the beam,* $P_C(\mathbf{x})$, *end deflection of the beam,* $\delta(\mathbf{x})$, *and side constraints. The four design variables,* h, l, t, *and* b, *are denoted as* x_1, x_2, x_3, *and* x_4, *respectively, as illustrated in Figure 10. The details of this practical optimal welded beam design problem are:*

$$f(\mathbf{x}) = (1 + c_1)x_1^2 x_2 + c_2 x_3 x_4 (x_2 + 14),$$

$$\textit{subject to} \quad \begin{array}{ll} g_1(\mathbf{x}) = \tau(\mathbf{x}) - \tau_{\max} \le 0, & g_2(\mathbf{x}) = \sigma(\mathbf{x}) - \sigma_{\max} \le 0, \\ g_3(\mathbf{x}) = x_1 - x_4 \le 0, & g_4(\mathbf{x}) = \delta(\mathbf{x}) - \delta_{\max} \le 0, \\ g_5(\mathbf{x}) = F - P_C(\mathbf{x}) \le 0, & \end{array}$$

where c_1 ($c_1 = 0.10471$) *and* c_2 ($c_2 = 0.04811$) *are the unit costs of the weld material and bar stock, respectively,* $\tau(\mathbf{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$, $\tau' = \frac{F}{\sqrt{2}x_1x_2}$, $\tau'' = \frac{MR}{J}$, $M = F(L + \frac{x_2}{2})$, $R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1+x_3}{2})^2}$, $J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + (\frac{x_1+x_3}{2})^2\right]\right\}$, $\sigma(\mathbf{x}) = \frac{6FL}{x_3^2x_4}$, $\delta(\mathbf{x}) = \frac{4FL^3}{Ex_3^3x_4}$, $P_C(\mathbf{x}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$, $F = 6000$ *lb*, $L = 14$ *in*, $\delta_{\max} = 0.25$ *in*, $E = 30\times10^6$ *psi*, $G = 12\times10^6$ *psi*, $\tau_{\max} = 13,600$ *psi*, $\sigma_{\max} = 30,000$ *psi*, $0.125 \le x_1 \le 5$, $0.1 \le x_2, x_3 \le 10$, *and* $0.1 \le x_4 \le 5$.

Again, we use the same parameters as in Example 5.1, except that NFE = 100,000, and β *is chosen to be* $\beta = 0.1$. *Table 6 shows the average optimization results of our HS-CA and HS method. We can observe that the former is capable of outperforming the later in attacking this demanding welded beam design problem.*

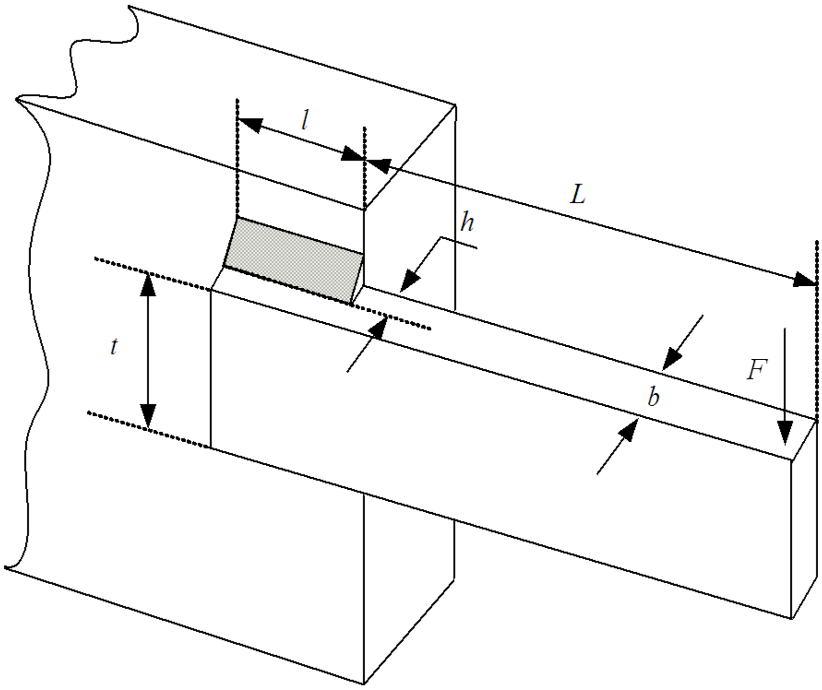


FIGURE 10. Welded beam design

TABLE 6. Optimization results of welded beam design with HS and HS-CA

	x_1	x_2	x_3	x_4	$f(\mathbf{x})$
HS	0.269560	4.114880	7.912975	0.305209	2.306814
HS-CA	0.299005	2.744191	7.502979	0.311244	2.093270

Example 5.3. *Optimal gear train design. The goal of the optimal gear train design is to minimize the cost of the Gear Ratio (GR) of a gear train, which is shown in Figure 11 [29]. The GR can be defined as:*

$$GR = \frac{n_B n_D}{n_F n_A}, \tag{19}$$

where n_A, n_B, n_D, *and* n_F *are the numbers of the teeth of the gearwheel in Figure 11, and they are denoted as* x_1, x_2, x_3, *and* x_4, *respectively in this engineering optimization problem.* x_1, x_2, x_3, *and* x_4 *are all integers between* $[12, 60]$. *The objective function* $f(\mathbf{x})$ *to be minimized is*

$$f(\mathbf{x}) = \left(\frac{1}{6.931} - \frac{x_3 x_2}{x_1 x_4}\right). \tag{20}$$

The same simulation parameters as in the above example are used in the HS and HS-CA here. The average *optimal design variables and costs acquired are given in Table 7, which actually demonstrate the superior optimization capability of our HS-CA.*

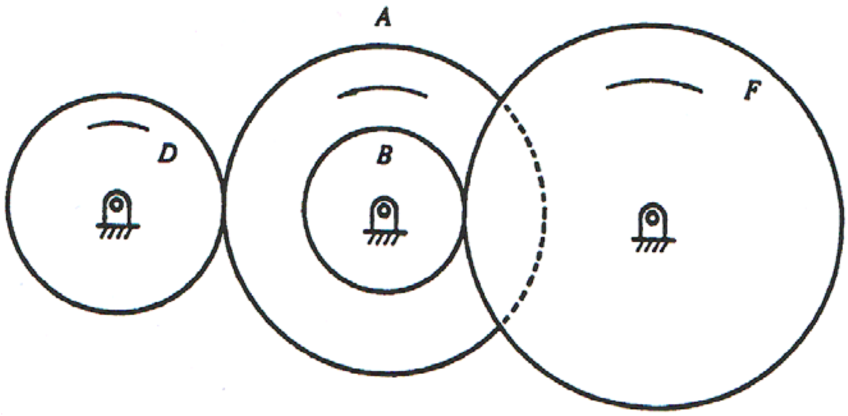


FIGURE 11. Gear train design

TABLE 7. Optimization results of gear train design with HS and HS-CA

	x_1	x_2	x_3	x_4	$f(\mathbf{x})$
HS	47.91	20.12	17.63	48.66	9.3078×10^{-10}
HS-CA	48.94	19.14	18.73	49.31	2.4815×10^{-10}

Example 5.4. *Optimal pressure vessel design. The structure of the center and end section of a pressure vessel is shown in Figure 12, which is made of carbon steel ASME SA 203 grade B [30]. The objective of the optimal design is to find a feasible set of dimensions T_s (shell thickness), T_h (spherical head thickness), R (radius of cylindrical shell), and L (shell length) with a minimum total manufacturing cost for the pressure vessel, subject to the constraints on the minimal wall thicknesses, the minimal value of the tank, as well as the length of the cylindrical shell. The design variables T_s, T_h, R, and L are denoted as x_1, x_2, x_3, and x_4, respectively here. Note that the variables T_s and T_h must be the integer multiples of 0.0625.*

$$\begin{aligned}
\textit{minimize:} \quad & f(\mathbf{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1611x_1^2x_4 + 19.84x_1^2x_3, \\
\textit{subject to:} \quad & g_1(\mathbf{x}) = 0.0193x_3 - x_1 \leq 0, \\
& g_2(\mathbf{x}) = 0.00954x_3 - x_2 \leq 0, \\
& g_3(\mathbf{x}) = 750 \times 1728 - \pi x_3^2 x_4 - 0.75\pi x_3^3 \leq 0, \\
& g_4(\mathbf{x}) = x_4 - 240 \leq 0, \\
& 1 \leq x_1, x_2 \leq 99, 10.0 \leq x_3, x_4 \leq 200.0.
\end{aligned}$$

The average optimized pressure vessel design variables and costs obtained by the HS and HS-CA are provided in Table 8. Apparently, the optimization performance of the proposed HS-CA is moderately better than that of the original HS method in manipulating this optimal design problem.

TABLE 8. Optimization results of pressure vessel design with HS and HS-CA

	x_1	x_2	x_3	x_4	$f(\mathbf{x})$
HS	1.8088	2.1281	57.0342	108.2850	2.3396×10^4
HS-CA	1.0781	1.0006	54.0931	69.1591	9.2040×10^3

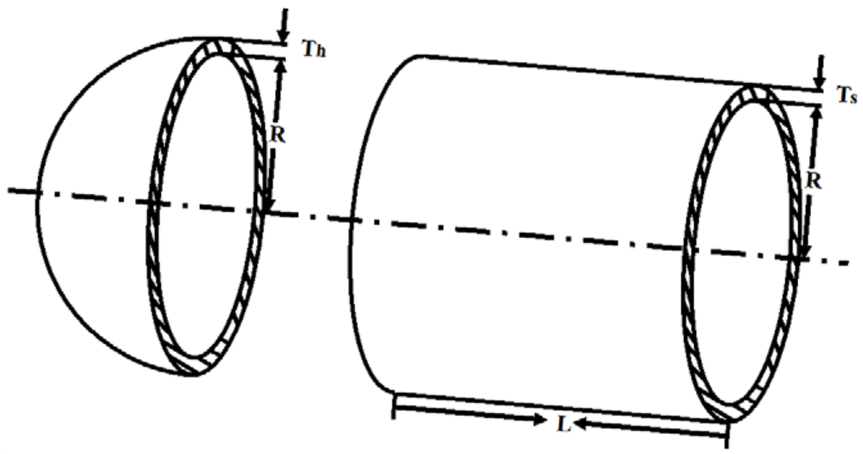


FIGURE 12. Pressure vessel design

6. Optimal Design of Permanent Magnet Direct Driven Wind Generator: A Case Study.

6.1. **Structure of wind generator.** The wind generator shown in Figures 13-15 is a radial flux type permanent magnet generator, in which the NdFeB magnets are surface mounted. 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 form and the constant dimensions of the slot are illustrated in Figure 14. The stator 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, the bearing shields, and the rotor steel body are all 20 mm thick (Figure 15). In the rotor body disc, there are holes, and around 50% of the disc is iron and 50% holes. The iron loss factor of the stator lamination is $p_{15} = 6.6$ W/kg with 50 Hz and 1.5 T. The air-gap length is 5 mm. The rated values of this generator are given in Table 9.

The stator winding overhang is presented in Figure 16. The constant dimensions are also shown in this figure. The clearance between the coils is 4 mm. The bending angle α

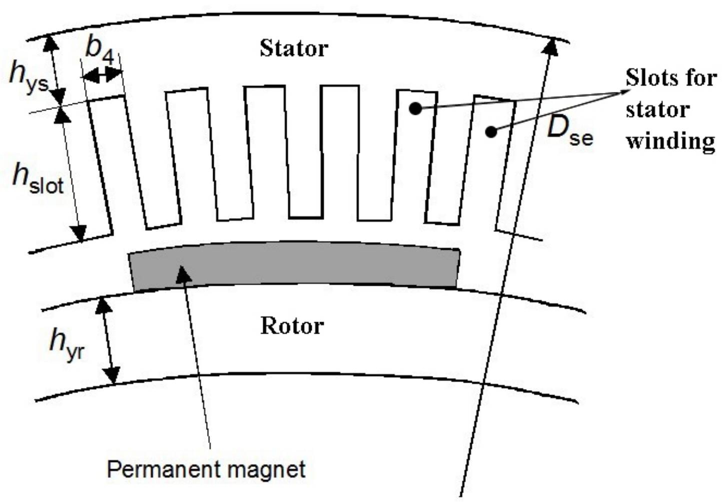


FIGURE 13. Cross-section and dimensions of permanent magnet generator

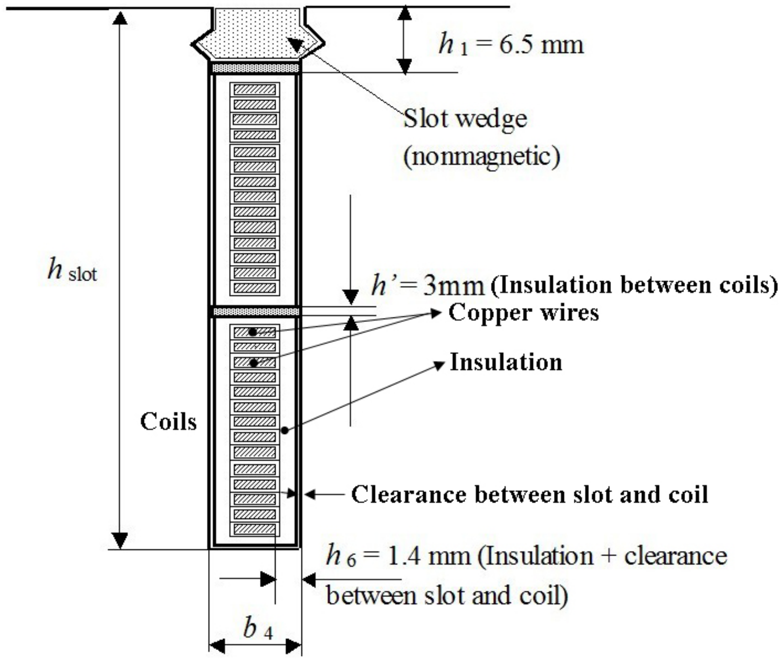


Figure 14. Stator slot form and constant dimensions of slot

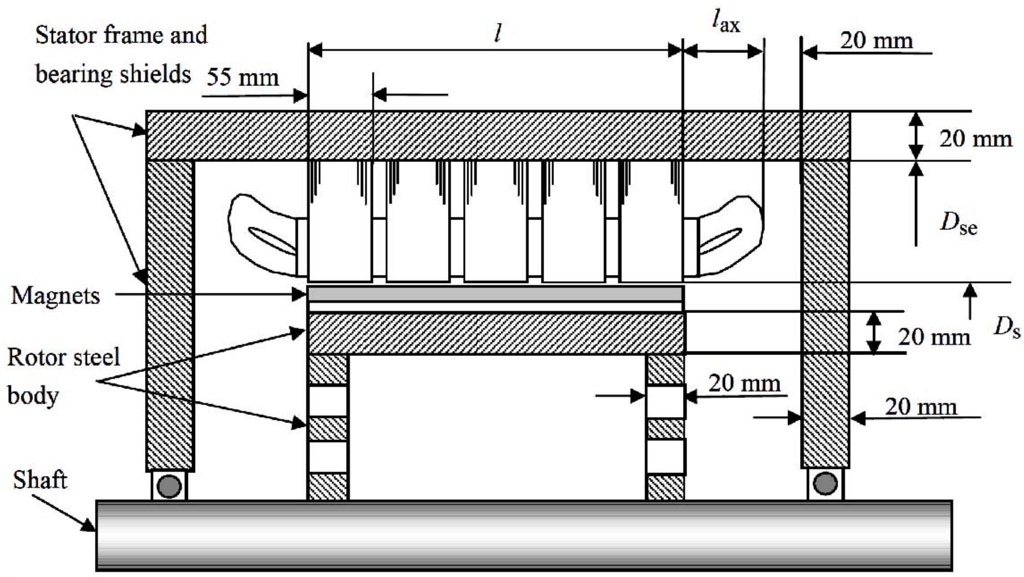


Figure 15. Axial cross-section of permanent magnet generator

Table 9. Rated values of generator

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

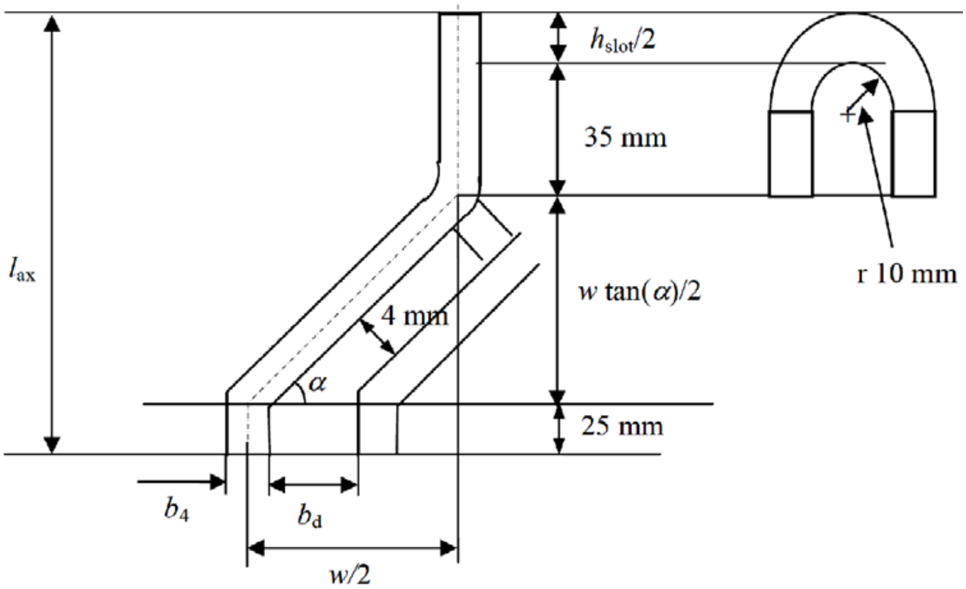


FIGURE 16. The overhang form and dimensions of the stator winding

is

$$\alpha = \arcsin \frac{b_4 + 4\ \text{mm}}{b_4 + b_\text{d}}, \tag{21}$$

where b_4 is the slot width, and b_d is the tooth width. The coil span at the middle of the slot is

$$w = \frac{\pi \left(D_\text{s} + h_\text{slot}\right)}{2p}, \tag{22}$$

where D_s is the air gap diameter, h_slot is the slot height, and p is the pole pair number. The axial length of the overhang is

$$l_\text{ax} = \frac{w \tan \alpha}{2} + \frac{h_\text{slot}}{2} + 60\ \text{mm}. \tag{23}$$

The average conductor length of a winding turn is

$$l_\text{av} = 2\left(l + \frac{w}{\cos \alpha} + \pi \frac{h_\text{slot}}{2} + 131\ \text{mm}\right), \tag{24}$$

where l is the length of the stator core (Figure 15).

6.2. **Objective function and design parameters.** The design principles of the above wind generator are given in [31]. 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 the generator:

$$f(\boldsymbol{x}) = k_\text{Fe} m_\text{Fe} + k_\text{Cu} m_\text{Cu} + k_\text{PM} m_\text{PM} + k_\text{Fef} m_\text{Frame} + k_\text{Loss} P_\text{tot}, \tag{25}$$

where m_Fe, m_Cu, m_PM, and m_Frame are the masses of the stator iron core, the stator winding, the permanent magnets, and the stator frame and rotor body, respectively, k_Fe, k_Cu, k_PM, and k_Fef are the unit prices of the stator core, the copper, the permanent magnets, and the stator frame and rotor body, respectively, and k_Loss is the capitalized costs of the losses (Table 10). The cost of the stator core actually includes the punching, the waste parts of the sheet, as well as the 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, the assembly into bigger cassettes, and the 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 10. Unit prices of materials and capitalized loss costs

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 °C, and the iron losses in the stator teeth are

$$P_{\mathrm{Fed}} = 2 \cdot p_{15} \left(B_{\mathrm{d}}/1.5\mathrm{T}\right)^2 \left(f/50\mathrm{Hz}\right)^{1.5} m_{\mathrm{d}}, \tag{26}$$

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

$$P_{\mathrm{Fey}} = 1.5 \cdot p_{15} \left(B_{\mathrm{y}}/1.5\mathrm{T}\right)^2 \left(f/50\mathrm{Hz}\right)^{1.5} m_{\mathrm{y}}, \tag{27}$$

where B_{y} is the maximum flux density in the yoke, and m_{y} is 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

$$P_{\rho} = 10 \cdot D_{\mathrm{r}} \left(l + 0.6 \cdot \tau_{\mathrm{p}}\right) \left(\pi n D_{\mathrm{r}}\right)^2 [\mathrm{W}], \tag{28}$$

where D_{r} is the outer rotor diameter ([m]), τ_{p} is the pole pitch ([m]), and n is the rotational frequency of the rotor ([1/s]) [32]. Table 11 gives the nine design parameters to be optimized and their valid ranges. A total of five practical constraints are also provided in Table 12.

TABLE 11. Design parameters with ranges

Parameters	Symbols	Ranges
Stator outer diameter	D_{se}	3.0-15.0 m
Stator core length including the ventilation ducts	l	0.2-3.0 m
Stator yoke height	h_{ys}	0.01-0.5 m
Rotor yoke height	h_{yr}	0.01-0.5 m
Stator slot height	h_{slot}	0.07-0.3 m
Stator slot width	b_4	0.007-0.04 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
Number of poles pairs	p	20-100

TABLE 12. Optimization constraints

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

7. **HS-CA-Based Optimal Wind Generator Design.** In this section, we investigate the effectiveness of the proposed HS-CA in the above optimal wind generator design problem. The HS coefficients are chosen as the same as in Section 5. The CA parameters of Top and β in the HS-CA are 10% and 0.5, respectively. After a total of 100 independent trials have been run, the average convergence procedures of the HS and four variants of our HS-CA within 1,000 and 10,000 NFE are illustrated in Figures 17 and 18, respectively. Figures 19 and 20 show respectively the corresponding optimal costs acquired by the HS and HS-CA during the 100 trials. Note that the costs in these two figures have been ranked. Especially, Tables 13 and 14 present the optimal wind generator parameters and costs obtained after 1,000 and 10,000 NFE, respectively. The best costs, worst costs, and average costs obtained in these two cases are also summarized in Tables 15 and 16.

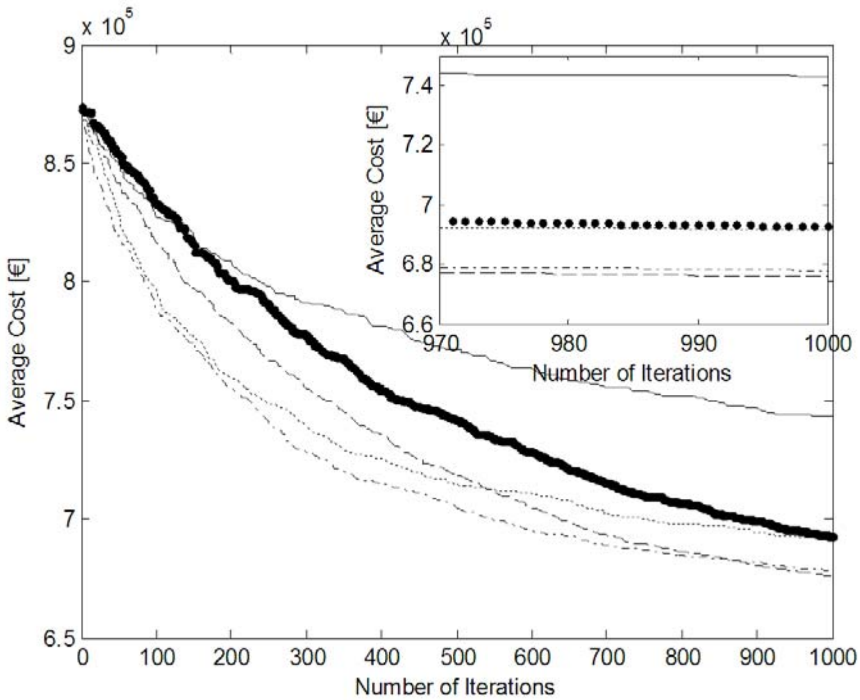


Figure 17. Average convergence procedures of HS and HS-CA within 1,000 NFE (thick line: HS, solid line: HS-CA (N_s), dotted line: HS-CA (S_d), dash-dot line: HS-CA ($N_s + S_d$), dashed line: HS-CA ($N_s + N_d$))

It is well known that the stator outer diameter, D_{se}, is one of the most important design parameters of our wind generator. Therefore, the relationship between the optimal costs and D_{se} is demonstrated in Figure 21. For each D_{se} in Figure 21, the optimal cost is chosen from 100 independent 10,000-NFE trials on the basis of HS-CA ($N_s + N_d$). We can observe that at the beginning, with D_{se} growing from 4 m, the cost acquired becomes smaller and smaller. The best one can be obtained, when D_{se} is approximately 8.2 m. Nevertheless, this relationship curve is rather flat. If the optimal outer diameter 8.2 m is halved to 4 m, the capitalized cost will increase by only around 15%.

As we know that the optimization performance of the CA heavily depends on its parameters, such as β. Different CA coefficient values applied can result in significant differences in the optimal solutions ultimately acquired. Figure 22 shows the relationship between the optimal costs obtained by the HS-CA ($N_s + N_d$) and varying β, when β grows from 0.1 to 1. Unfortunately, β is usually an application dependent parameter, and there is

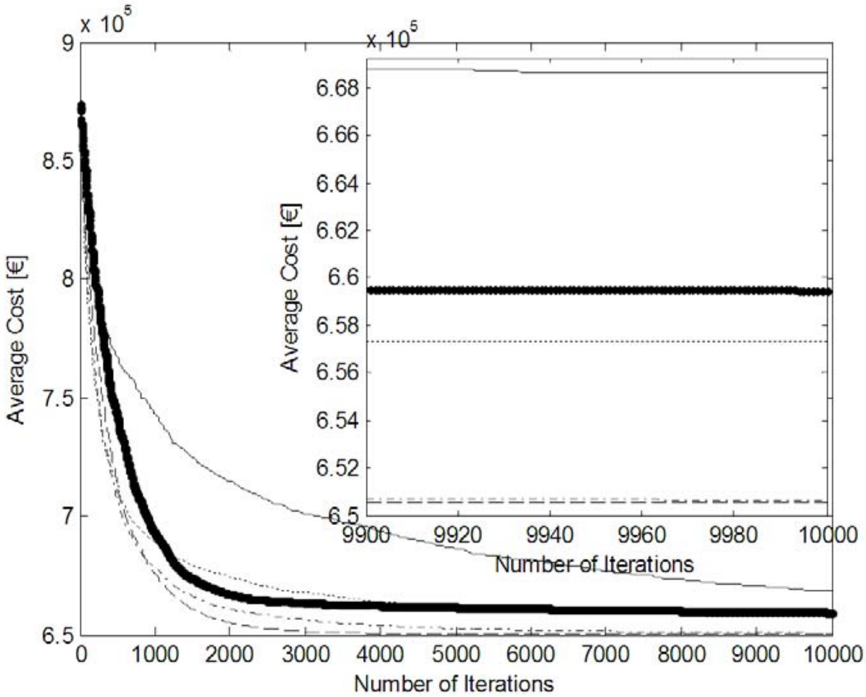


FIGURE 18. Average convergence procedures of HS and HS-CA within 10,000 NFE (thick line: HS, solid line: HS-CA (N_s), dotted line: HS-CA (S_d), dash-dot line: HS-CA ($N_s + S_d$), dashed line: HS-CA ($N_s + N_d$))

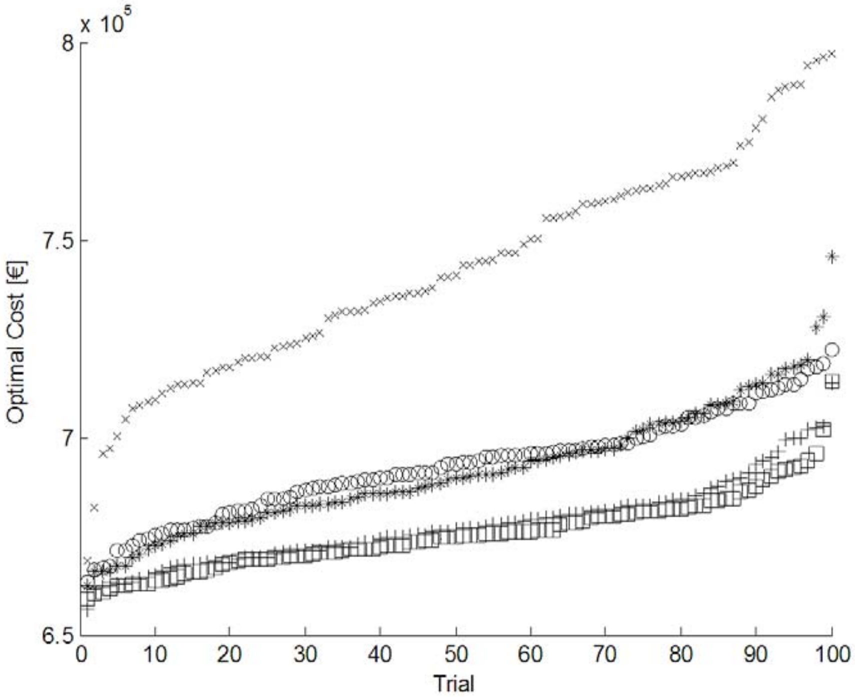


FIGURE 19. Optimal costs acquired by HS and HS-CA within 1,000 NFE. ('circle': HS, 'x-mark': HS-CA (N_s), 'star': HS-CA (S_d), 'plus': HS-CA ($N_s + S_d$), 'square': HS-CA ($N_s + N_d$)).

TABLE 13. Optimal parameters and costs acquired by HS and HS-CA within 1,000 NFE

	HS	HS-CA (N_s)	HS-CA (S_d)	HS-CA ($N_s + S_d$)	HS-CA ($N_s + N_d$)
l	0.4548	0.5288	0.5471	0.4687	0.4721
h_{ys}	0.0666	0.0539	0.0343	0.0416	0.0334
D_{se}	8.1120	7.9314	8.4834	8.1119	8.1357
h_{slot}	0.1137	0.0965	0.0991	0.1189	0.0920
$B_{\max}$	0.7057	0.7079	0.6188	0.6914	0.6613
z_{s}	22	22	22	20	22
h_{yr}	0.0351	0.0225	0.0248	0.0294	0.0455
p	70	82	95	76	75
b_4	0.0153	0.0102	0.0110	0.0123	0.0144
Costs	6.6350×10^5	6.6869×10^5	6.6243×10^5	6.5630×10^5	6.5917×10^5

TABLE 14. Optimal parameters and costs acquired by HS and HS-CA within 10,000 NFE

	HS	HS-CA (N_s)	HS-CA (S_d)	HS-CA ($N_s + S_d$)	HS-CA ($N_s + N_d$)
l	0.4691	0.4436	0.4569	0.4719	0.4785
h_{ys}	0.0386	0.0455	0.0396	0.0387	0.0380
D_{se}	8.1802	8.6117	8.6409	8.3130	8.2067
h_{slot}	0.1021	0.1034	0.0990	0.1002	0.0999
$B_{\max}$	0.6900	0.6606	0.6722	0.6735	0.6779
z_{s}	22	22	22	22	22
h_{yr}	0.0288	0.0306	0.0221	0.0260	0.0260
p	75	72	95	78	77
b_4	0.0134	0.0139	0.0113	0.0133	0.0134
Costs	6.5029×10^5	6.5366×10^5	6.5286×10^5	6.4993×10^5	6.4988×10^5

TABLE 15. Best, worst, and average costs acquired by HS and HS-CA within 1,000 NFE

	HS	HS-CA (N_s)	HS-CA (S_d)	HS-CA ($N_s + S_d$)	HS-CA ($N_s + N_d$)
Best Costs	6.6350×10^5	6.6869×10^5	6.6243×10^5	6.5630×10^5	6.5917×10^5
Worst Costs	7.2213×10^5	7.9717×10^5	7.4591×10^5	7.1400×10^5	7.1417×10^5
Average Costs	6.9266×10^5	7.4301×10^5	6.9166×10^5	6.7784×10^5	6.7574×10^5

TABLE 16. Best, worst, and average costs acquired by HS and HS-CA within 10,000 NFE

	HS	HS-CA (N_s)	HS-CA (S_d)	HS-CA ($N_s + S_d$)	HS-CA ($N_s + N_d$)
Best Cost	6.5029×10^5	6.5366×10^5	6.5286×10^5	6.4993×10^5	6.4988×10^5
Worst Cost	6.7621×10^5	6.8656×10^5	6.6497×10^5	6.5182×10^5	6.5418×10^5
Average Costs	6.5947×10^5	6.6862×10^5	6.5733×10^5	6.5065×10^5	6.5052×10^5

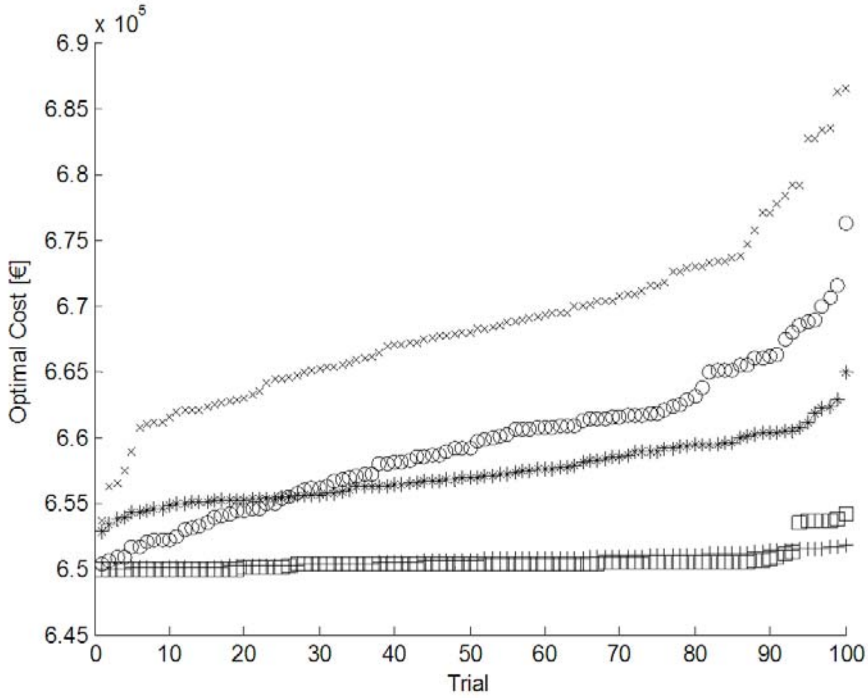


Figure 20. Optimal costs acquired by HS and HS-CA within 10,000 NFE ('circle': HS, 'x-mark': HS-CA (N_s), 'star': HS-CA (S_d), 'plus': HS-CA ($N_s + S_d$), 'square': HS-CA ($N_s + N_d$))

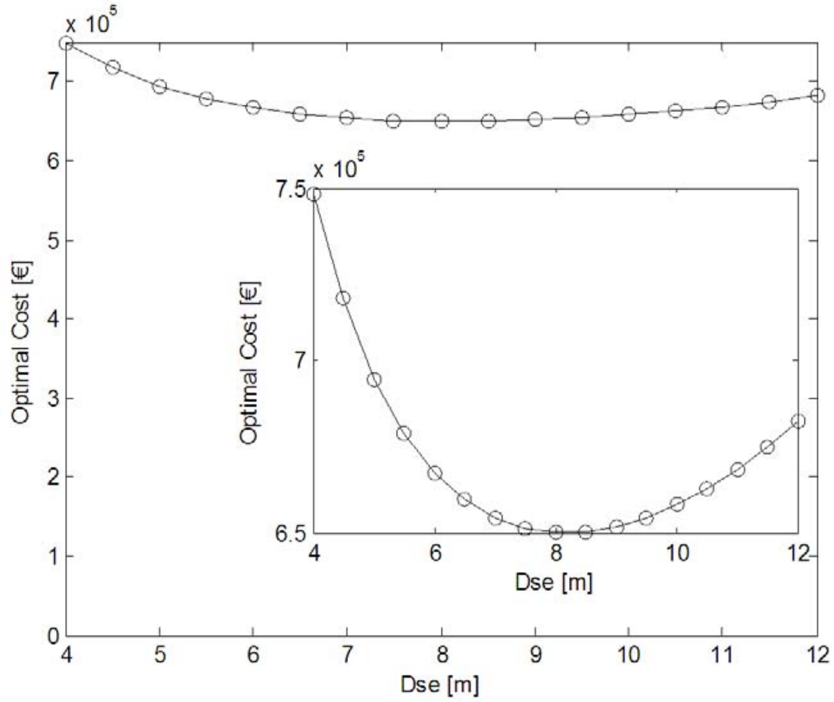


Figure 21. Relationship between optimal costs and stator outer diameter (D_{se})

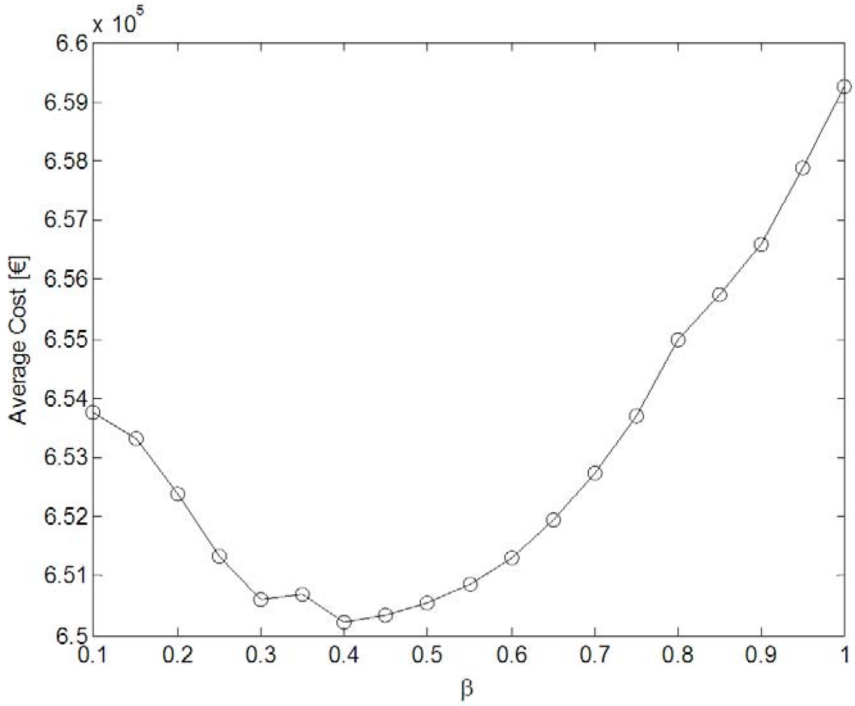


FIGURE 22. Relationship between optimal costs and parameter β in HS-CA ($N_s + N_d$)

no analytic way yet to choose its best value for a specific problem. Thus, it is generally determined based on a *trial and error* procedure, which might be time-consuming in practice.

Apparently, compared with the original HS method, the three variants of our HS-CA (HS-CA (S_d), HS-CA ($N_s + S_d$), and HS-CA ($N_s + N_d$)) can achieve moderately better average optimization results within the same NFE, because of the CA-based utilization of the search knowledge extracted from the HS. For example, HS-CA ($N_s + N_d$) offers improvements of about 2.4% and 1.4% in the average optimized cost in Tables 15 and 16, respectively. Nevertheless, the HS-CA has a higher computation complexity than that of the HS algorithm, due to the incorporated CA operations. Furthermore, as aforementioned, the performance of the HS-CA is indeed affected by the *Influence* function used, which plays a pivotal role in its optimization capability. As a matter of fact, from Tables 15 and 16, we can find out that HS-CA (N_s) is even worse than the original HS method. To summarize, in this optimal design of wind generator problem, the optimization capabilities of the four HS-CA variants can be ranked as follows: HS-CA ($N_s + N_d$) >HS-CA ($N_s + S_d$) >HS-CA (S_d) >HS-CA (N_s).

8. **Conclusions.** In this paper, we propose a new hybrid optimization approach by merging the HS method and CA together. The belief space of the CA is updated based on the information provided by the HS. On the other hand, the mutation operation of the HS is efficiently controlled by the situational knowledge and normative knowledge from the CA. Some simulation examples of nonlinear functions and engineering optimization using the HS-CA have been demonstrated. An optimal wind generator design problem is also deployed to further verify the effectiveness of the proposed method. It can be observed that the employment of the CA in the HS-CA indeed yields an improved convergence performance. Compared with the original HS, better optimization results are obtained by

the HS-CA. Therefore, our HS-CA has promising potentials in coping with a large variety of engineering optimization problems.

Actually, the 'No Free Lunch' theorem is a fundamental barrier to the exaggerated claims of the power and efficiency of any specific optimization algorithm [33]. In other words, there is no single optimization method that can be the best for all kinds of engineering problems. One possible way to handle the negative implication of this 'No Free Lunch' theorem is to restrict the applications of a given algorithm to only a particular type of tasks. Therefore, we are going to further study the feasibility and applicability of the HS-CA. In addition, the CA parameters and $Influence$ function have been proved to have a critical effect on the behavior of our HS-CA. The appropriate selection of these factors to deal with more real-world problems can be another investigation topic for the future work.

Acknowledgment. This research work was funded by the project FOVI that is part of FIMECC-program EFFIMA and Academy of Finland under Grants 135225, 127299 and 137837. 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] 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.
[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] O. M. Alia, R. Mandava and M. E. Aziz, A hybrid harmony search algorithm for MRI brain segmentation, *Evolutionary Intelligence*, vol.4, no.1, pp.31-49, 2011.
[6] N. Nahas and D. Thien-My, Harmony search algorithm: Application to the redundancy optimization problem, *Engineering Optimization*, vol.42, no.9, pp.845-861, 2010.
[7] 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.
[8] M. G. H. Omran and M. Mahdavi, Global-best harmony search, *Applied Mathematics and Computation*, vol.198, no.2, pp.643-656, 2008.
[9] 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.10(A), pp.2985-2996, 2009.
[10] 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.
[11] 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.
[12] 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 Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol.41, no.1, pp.89-106, 2011.
[13] R. G. Reynolds and B. Peng, Cultural algorithms: Modeling how cultures learn to solve problems, *Proc. of the IEEE International Conference on Tools with Artificial Intelligence*, Boca Raton, FL, pp.166-173, 2004.
[14] R. G. Reynolds and C. J. Chung, CAEP: An evolution-based tool for real-valued function optimization using cultural algorithms, *International Journal on Artificial Intelligence Tools*, vol.7, no.3, pp.239-293, 1998.
[15] R. Poli and W. B. Langdon, *Foundations of Genetic Programming*, Springer, Berlin, Germany, 2002.

[16] M. Krug, S. K. Nguang, J. Wu and J. Shen, GA-based model predictive control of boiler-turbine systems, *International Journal of Innovative Computing, Information and Control*, vol.6, no.11, pp.5237-5248, 2010.
[17] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, John Wiley & Sons Ltd., West Sussex, England, 2005.
[18] C.-J. Lin, J.-G. Wang and S.-M. Chen, 2D/3D face recognition using neural network based on hybrid Taguchi-particle swarm optimization, *International Journal of Innovative Computing, Information and Control*, vol.7, no.2, pp.537-553, 2011.
[19] 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.
[20] 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, pp.341-359, 1997.
[21] V. Vegh, G. K. Pierens and Q. M. Tieng, A variant of differential evolution for discrete optimization problems requiring mutually distinct variables, *International Journal of Innovative Computing, Information and Control*, vol.7, no.2, pp.897-914, 2011.
[22] R. G. Reynolds and C. J. Chung, Knowledge-based self-adaptation in evolutionary programming using cultural algorithms, *Proc. of the IEEE International Conference on Evolutionary Computation*, Indianapolis, IN, pp.71-76, 1997.
[23] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
[24] O. Kramer, Evolutionary self-adaptation: A survey of operators and strategy parameters, *Evolutionary Intelligence*, vol.3, no.2, pp.51-65, 2010.
[25] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd Edition, Springer, Berlin, Germany, 1996.
[26] *Benchmark Functions*, http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm.
[27] 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.
[28] K. Deb, Optimal design of a welded beam via genetic algorithms, *Journal of American Institute of Aeronautics and Astronautics*, vol.29, no.11, pp.2013-2015, 1991.
[29] K. E. Parsopoulos and M. N. Vrahatis, Unified particle swarm optimization for solving constrained engineering optimization problems, *Lecture Notes in Computer Science*, vol.3612, pp.582-591, 2005.
[30] C. A. C. Coello, Constraint-handling using an evolutionary multiobjective optimization technique, *Civil Engineering and Environmental Systems*, vol.17, no.4, pp.319-346, 2000.
[31] J. Pyrhönen, T. Jokinen and V. Hrabovcová, *Design of Rotating Electrical Machines*, John Wiley & Sons Ltd., West Sussex, UK, 2008.
[32] X. Z. Gao, T. Jokinen, X. Wang, S. J. Ovaska and A. Arkkio, A new harmony search method in optimal wind generator design, *Proc. of the XIX International Conference on Electrical Machines*, Rome, Italy, 2010.
[33] D. H. Wolpert and W. G. Macready, No free lunch theorems for optimization, *IEEE Trans. on Evolutionary Computation*, vol.1, no.1, pp.67-82, 1997.