Taylor & Francis
Taylor & Francis Group

Applied Artificial Intelligence
An International Journal

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

A Hybrid Greedy Sine Cosine Algorithm with Differential Evolution for Global Optimization and Cylindricity Error Evaluation

Qijun Li, Huifeng Ning, Jun Gong, Xiao Li & Baolin Dai

To cite this article: Qijun Li, Huifeng Ning, Jun Gong, Xiao Li & Baolin Dai (2021) A Hybrid Greedy Sine Cosine Algorithm with Differential Evolution for Global Optimization and Cylindricity Error Evaluation, Applied Artificial Intelligence, 35:2, 171-191, DOI: 10.1080/08839514.2020.1848276

To link to this article: https://doi.org/10.1080/08839514.2020.1848276

Published online: 15 Dec 2020.

APPLIED ARTIFICIAL INTELLIGENCE
2021, VOL. 35, NO. 2, 171–191
https://doi.org/10.1080/08839514.2020.1848276

Taylor & Francis
Taylor & Francis Group

A Hybrid Greedy Sine Cosine Algorithm with Differential Evolution for Global Optimization and Cylindricity Error Evaluation

Qijun Li[a,b], Huifeng Ning[a], Jun Gong[a], Xiao Li[a], and Baolin Dai[a]

[a]School of Mechanical and Electrical Engineering, Lanzhou University of Technology, Lanzhou, China; [b]School of Electromechanics and Automobile Engineering, Tianshui Nomal University, Tianshui, China

ABSTRACT

Sine-cosine algorithm (SCA) has found a widespread application in various engineering optimization problems. However, SCA suffers from premature convergence and insufficient exploitation. Cylindricity error evaluation is a typical engineering optimization problem related to the quality of cylindrical parts. A hybrid greedy sine-cosine algorithm with differential evolution (HGSCADE) is developed in this paper to solve optimization problems and evaluate cylindricity error. HGSCADE integrates the SCA with the opposition-based population initialization, the greedy search, the differential evolution (DE), the success history-based parameter adaptation, and the Levy flight-based local search. HGSCADE is tested on the CEC2014 benchmark functions and is employed in cylindricity error evaluation. The results show the superiority of HGSCADE to other state-of-the-art algorithms for the benchmark functions and cylindricity error evaluation.

ARTICLE HISTORY
Received 6 September 2020
Revised 1 November 2020
Accepted 6 November 2020

Introduction

Optimization refers to reaching the optimal objective value of a certain problem under-related constraints. Optimization problems exist in different areas of engineering or science. Meta-heuristic algorithms (MAs) have been developed to solve complex optimization problems that possess the properties of discontinuity, nondifferentiability, nonconvexity, nonlinearity, and multimodality (Yang 2008). Generally, MAs take the optimization problem as a black box and search for the solution based on stochastic rules. By transferring and exchanging information among search agents, the positions of the search agents are constantly updated to a promising region with the optimal solution.

No Free Lunch (NFL) theorem (Wolpert and Macready 1997) states that it is impossible to solve all optimization problems by a specific algorithm, i.e., an algorithm is applicable to a given optimization problem that may not be applicable to another one with distinct characteristics. Therefore, MAs need to be further studied to deal with different optimization problems. The research

CONTACT Jun Gong gongjjdxy@sohu.com School of Mechanical and Electrical Engineering, Lanzhou University of Technology, Lanzhou, Gansu 730050, China.

directions of MAs can be categorized as initiating new algorithms, improving existing algorithms and combining different algorithms. Some of the well-known MAs are genetic algorithms (GAs) (Srinivas and Patnaik 1994), particle swarm optimization (PSO) (Kennedy and Eberhart 1995), differential evolution (DE) (Storn and Price 1997), ant colony optimization (ACO) (Dorigo and Di Caro 1999), harmony search (HS) (Geem, Kim, and Loganathan 2001), bacterial foraging optimization (BFO) (Passino 2002), artificial bee colony (ABC) (Karaboga and Basturk 2007), biogeography-based optimization (BBO) (Simon 2008), gravitational search algorithm (GSA) (Rashedi, Nezamabadi-pour, and Saryazdi 2009), gray wolf optimizer (GWO) (Mirjalili, Mirjalili, and Lewis 2014), ant lion optimizer (ALO) (Mirjalili 2015), whale optimization algorithm (WOA) (Mirjalili and Lewis 2016), sine-cosine algorithm (SCA) (Mirjalili 2016), salp swarm algorithm (SSA) (Mirjalili et al. 2017).

As a crucial index to evaluate the machining accuracy and form error of cylindrical parts, cylindricity error is closely related to the product quality and performance. The minimum zone cylindricity (MZC) error evaluation method is consistent with the international standards, but a general algorithm that corresponds to MZC has not been developed. Essentially, cylindricity error evaluation can be regarded as a single objective real-parameter numerical optimization problem. Therefore, MAs has been applied for cylindricity error evaluation under the MZC principle, such as GA (Lai et al. 2000), PSO (Wen et al. 2010), HS (Yang et al. 2018), etc. It is very important to improve the convergence speed and accuracy of MAs in cylindricity error evaluation.

SCA, proposed by Mirjalili (2016), is a population-based optimization algorithm for solving optimization problems. SCA presents a good performance in some cases (Das, Bhattacharya, and Chakraborty 2018; Li, Fang, and Liu 2018; Mirjalili 2016), but it suffers from premature convergence and insufficient exploitation. Therefore, SCA variants were developed and applied for different optimization problems like feature selection (Sindhu et al. 2017), object tracking (Nenavath and Jatoth 2018; Nenavath, Kumar Jatoth, and Das 2018), pairwise local sequence alignment (Issa et al. 2018), optimal power flow (Attia, El Sehiemy, and Hasanien 2018), engineering design (Abd Elaziz, Oliva, and Xiong 2017; Chegini, Bagheri, and Najafi 2018; Gupta and Deep 2019; Rizk-Allah 2018), etc.

In this study, a novel algorithm called hybrid greedy sine-cosine algorithm with differential evolution (HGSCADE) is proposed for solving benchmark problems and evaluating cylindricity error. In HGSCADE, the opposition-based learning (OBL) is applied in population initialization, and the greedy search is employed to prevent excluding good solutions. Meanwhile, the DE is incorporated into SCA to improve the exploitation ability and maintain the population diversity, and a success history-based parameter adaptation is introduced to overcome invalid search. Then, a Levy flight-based local search is used to escape from local optima. HGSCADE is compared with standard SCA, other state-of-the-art MAs, and SCA variants over the CEC2014 benchmark functions. Finally, the proposed

HGSCADE algorithm is used for cylindricity error evaluation as a real engineering case. The test and experiment results testify to the superiority and effectiveness of the proposed algorithm.

The remainder of this paper is organized as follows: the brief introduction of SCA is present in Section 2. The proposed HGSCADE algorithm is described in Section 3. The test results and comparisons are presented in Section 4. The application of HGSCADE for cylindricity error evaluation is introduced in Section 5. Eventually, conclusions are drawn in Section 6.

Sine-Cosine Algorithm

Similar to other MAs, the standard SCA starts with random candidate agents. Then, the agents are updated by sine and cosine functions to create new ones.

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^{t} + r_1 \times \sin(r_2) \times \left| r_3 X_{gb,j}^{t} - X_{i,j}^{t} \right|, & r_4 < 0.5 \\ X_{i,j}^{t} + r_1 \times \cos(r_2) \times \left| r_3 X_{gb,j}^{t} - X_{i,j}^{t} \right|, & r_4 \geq 0.5 \end{cases} \tag{1}$$

where $X_{i,j}^{t}$ is the position of the i-th agent in j-th dimension at t-th iteration, $r_1/r_2/r_3$ are random numbers, $X_{gb,j}^{t}$ is the best position obtained so far in j-th dimension, r_4 is a random number in [0, 1], and || denotes the absolute value.

The variable r_1 is adjusted with the number of iterations by Equation (2):

$$r_1 = a - (a \times t)/T \tag{2}$$

where a is a constant, t is the t-th iteration and T represents the upper limit of iterations.

The Proposed Algorithm

In this section, the proposed HGSCADE algorithm is introduced in detail. Although the standard SCA has found a widespread application in various engineering optimization problems, it suffers from premature convergence and insufficient exploitation. To overcome the drawbacks of SCA, we integrate five strategies into SCA as follows:

Opposition-based Population Initialization

Random population initialization generates some useless agents due to the absence of systemic knowledge. Therefore, the OBL is employed to form the initial population. The opposite value of a real number $x \in [l, u]$ $(l, u \in R)$ is specified as (Tizhoosh 2005):

$$\bar{x} = l + u - x \tag{3}$$

where l and u are the lower limit and upper limit of x, respectively.

Firstly, a population $\boldsymbol{X}^0 = \{\boldsymbol{X}_i^0 = [x_{i,1}^0, x_{i,2}^0, \ldots, x_{i,d}^0]\}_{i=1}^N$ is generated from the uniform distribution. Secondly, the opposite vector $\bar{\boldsymbol{X}}_i^0 (i = 1, 2, \ldots, N)$ for each agent is calculated by Equation (3). Thirdly, the objective values of the populations $\boldsymbol{X}^0$ and $\bar{\boldsymbol{X}}^0$ are computed, and the best N agents are selected from $\boldsymbol{X}^0 \cup \bar{\boldsymbol{X}}^0$, as the initial population.

Greedy Search

In the earlier search stage of SCA ($r_1 > 1$), the optimal position might be skipped over. Therefore, the greedy search is employed to update the position of each agent, i.e.

$$\boldsymbol{X}_i^{t+1} = \begin{cases} \boldsymbol{X}_i^{t+1}, & fit(\boldsymbol{X}_i^{t+1}) < fit(\boldsymbol{X}_i^t) \\ \boldsymbol{X}_i^t, & otherwise \end{cases}, \quad i = 1, 2, \ldots, N \tag{4}$$

where $\mathbf{X}_i^t$ and $\mathbf{X}_i^{t+1}$ are the parent position and new position of i-th agent, respectively.

Combined with Differential Evolution

The standard SCA uses the knowledge of a best-so-far agents to produce new members, which may result in insufficient population diversity and premature convergence. Hence, DE is incorporated into SCA in a parallel way. The component of position oversteps the boundary is set to be the mean of the overstepped boundary and the corresponding component of the parent position. For each iteration, the agent i will be updated by the SCA method if $rand_i < pu_i$, where $rand_i$ is a random number in [0, 1] and $pu_i \in [0, 1]$ is the updating rate. Otherwise, the agent i will be updated by the DE method as follows:

Firstly, a donor/mutant vector $\boldsymbol{V}_i^t$ is produced by the parent vectors$\{\boldsymbol{X}_i\}_{i=1}^N$.

$$\boldsymbol{V}_i^t = \begin{cases} \boldsymbol{X}_{R_1}^t + F_i \cdot (\boldsymbol{X}_{R_2}^t - \boldsymbol{X}_{R_3}^t), & rand_m < p_m \\ \boldsymbol{X}_i^t + F_i \cdot (\boldsymbol{X}_{pt}^t - \boldsymbol{X}_i^t) + F_i \cdot (\boldsymbol{X}_{R_2}^t - \boldsymbol{X}_{R_3}^t), & rand_m \geq p_m \end{cases} \tag{5}$$

where indexes R_1, R_2, and R_3 represent integers stochastically generated from [1, N], and $R_1 \neq R_2 \neq R_3 \neq i$. F_i is the scaling factor uniformly distributed in [0, 1]. $\boldsymbol{X}_{pt}^t$is the position of the one picked out from the top $N \times pt$ ($pt \in (0, 1)$) agents in the population at t-th iteration stochastically (the agents in the population are sorted according to their fitness values), $rand_m$ represents a random number in [0, 1], $p_m = 1 - t/T$.

Secondly, a trial vector $\boldsymbol{U}_i^t$ is generated by crossing $\boldsymbol{V}_i^t$ with $\boldsymbol{X}_i^t$.

$$U_{i,j}^{t}=\begin{cases} V_{i,j}^{t}, & r_{ij}<cr \text{ or } j=j_0 \\ X_{i,j}^{t}, & otherwise \end{cases}, \; j=1,2,\ldots,d \tag{6}$$

where r_{ij} is a random number uniformly distributed in [0, 1], j_0 is a random integer generated from [1, d], $cr \in$ [0, 1] denotes the crossover rate.

Finally, the survivor is selected from the better one between $\boldsymbol{U}_i^t$ and $\boldsymbol{X}_i^t$.

$$\boldsymbol{X}_i^{t+1}=\begin{cases} \boldsymbol{U}_i^t, & fit(\boldsymbol{U}_i^t)<fit(\boldsymbol{X}_i^t) \\ \boldsymbol{X}_i^t, & otherwise \end{cases}, \; i=1,2,\ldots,N \tag{7}$$

Success History-Based Parameter Adaptation

To reduce invalid search, HGSCADE regulates the critical parameters adaptively by the mechanism developed in (Tanabe and Fukunaga 2013). As shown in Table 1, the historical information for three memory parameters M_{pu}, M_F , and M_{cr} is stored with H entries. At each iteration, pu_i, F_i , and cr_i for each agent i are generated as follows:

$$\begin{bmatrix} pu_i \\ F_i \\ cr_i \end{bmatrix} = \begin{bmatrix} randn_i(M_{pu,r_i},0.1) \\ randc_i(M_{F,r_i},0.1) \\ randn_i(M_{cr,r_i},0.1) \end{bmatrix} \tag{8}$$

where index r_i is randomly selected from [1, H], $randn_i(\mu, \sigma)$ and $randc_i(\mu, \sigma)$ denote two values generated from Gaussian distribution and Cauchy distribution, respectively. In the case that the generated value of pu_i or cr_i is out of [0, 1], set it to the closest limit value. For F_i, truncate it to be 1 if $F_i \geq 1$ or regenerate it if $F_i < 1$.

At each iteration, the values of pu_i, F_i , and cr_i that successfully generate a superior offspring are recorded as S_{pu}, S_F , and S_{cr}. If and only if superior offspring are generated, the values of M_{pu}, M_F , and M_{cr} are updated as follows:

$$\begin{bmatrix} M_{pu,k} \\ M_{F,k} \\ M_{cr,k} \end{bmatrix} = \begin{bmatrix} mean_{WA}(S_{pu}) \\ mean_{WL}(S_F) \\ mean_{WA}(S_{cr}) \end{bmatrix} \tag{9}$$

where index k ($1 \leq k \leq H$) indicates the entry to be updated. k is initialized to 1 at the first iteration and is incremental with iteration. When $k > H$, reset k = 1. $mean_{WA}(S_{pu})$ and $mean_{WA}(S_{cr})$ are the weighted mean. $mean_{WL}(S_F)$ is the weighted Lehmer mean.

Table 1. The historical memory.

Index	1	2	…	H–1	H
M_{pu}	$M_{pu,1}$	$M_{pu,2}$	…	$M_{pu,H-1}$	$M_{pu,H}$
M_F	$M_{F,1}$	$M_{F,2}$	…	$M_{F,H-1}$	$M_{F,H}$
M_{cr}	$M_{cr,1}$	$M_{cr,2}$	…	$M_{cr,H-1}$	$M_{cr,H}$

$$\begin{bmatrix} mean_{WA}(S_{pu}) \\ mean_{WA}(S_{cr}) \\ mean_{WL}(S_F) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{|S_{pu}|} \omega_i \cdot S_{pu,i} \\ \sum_{j=1}^{|S_{cr}|} \omega_j \cdot S_{cr,j} \\ \left(\sum_{j=1}^{|S_{cr}|} \omega_j \cdot (S_{F,j})^2\right) \Big/ \left(\sum_{j=1}^{|S_{cr}|} \omega_j \cdot S_{F,j}\right) \end{bmatrix} \tag{10}$$

where $\omega_i = \Delta f_i \Big/ \sum_{i=1}^{|S_{pu}|} \Delta f_i$, $\omega_j = \Delta f_j \Big/ \sum_{j=1}^{|S_{cr}|} \Delta f_j$, $\Delta f_i = \left|f_i^{t+1}(\boldsymbol{X}_i) - f_i^t(\boldsymbol{X}_i)\right|$, $\Delta f_j = \left|f_j^t(\boldsymbol{U}_j) - f_j^t(\boldsymbol{X}_j)\right|$.

Local Search Based on Levy Flight

SCA may stagnate if the best agent is caught in local optima. Therefore, a Levy flight disturbance (Jensi and Jiji 2016) is applied to the best agent in the current population:

$$\boldsymbol{X}_{cbLevy} = \boldsymbol{X}_{cb} + step \oplus \mathrm{random}(\mathrm{size}(\boldsymbol{X}_{cb})) \tag{11}$$

where $step = Levy(\beta) \oplus \boldsymbol{X}_{cb}$, the symbol $\oplus$ denotes entry-wise multiplication. $Levy(\beta)$ is derived from Equation (12):

$$Levy(\beta) = scale \times \left(u \Big/ |v|^{1/\beta}\right) \tag{12}$$

where $scale$ is a factor to control the step size of Levy flight, $\beta \in [0, 2]$ is Levy index, u and v are derived from Gaussian distributions, i.e.

$$u \sim N(0, \sigma_u^2), \quad v \sim N(0, \sigma_v^2) \tag{13}$$

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[(1+\beta)/2]\beta 2^{(\beta-1)/2}} \right\}^{1/\beta}, \quad \sigma_v = 1 \tag{14}$$

where Γ is the standard gamma function.

Then, compute the opposite vector $\bar{\boldsymbol{X}}_*$ for $\boldsymbol{X}_{cbLevy}$, and substitute $\boldsymbol{X}_{cb}$ with the best one chosen from $\left\{\boldsymbol{X}_{cb}, \mathbf{X}_{cbLevy}, \bar{\boldsymbol{X}}_*\right\}$.

Overall Implementation

The overall implementation steps of HGSCADE are described in Algorithm 1. HGSCADE involves three stages: Firstly, the OBL is employed to form the initial population, and the initial values of parameters are given. Secondly, each agent in the population is updated by the greedy search SCA method or DE method according to the updating rate, and the success history-based mechanism is used for critical parameter adaptation. Thirdly, the Levy flight-based local search is applied to the current best agent, and the search continues until the termination condition is reached.

Algorithm 1 The HGSCADE

```
Randomly generate a population $\boldsymbol{X}$ of $N$ agents.
Compute the opposite population $\bar{\boldsymbol{X}}$ by Equation (3).
Evaluate the fitness of $\boldsymbol{X}$ $and$ $\bar{\boldsymbol{X}}$ by calculating the corresponding objective function values.
Select the best $N$ agents from $\boldsymbol{X} \cup \bar{\boldsymbol{X}}$
Initialize the values of parameters $a, M_{pu}, M_F, M_{cr}, pt, H, \beta, scale$
$k = 1, t = 0$
While ($t < T$) do
    Update the global best position $\boldsymbol{X}_{gb}$
    Randomly select an index $r_i$ from $[1, H]$
    Generate $pu$, $F$ and $cr$ by Equation (8)
    For ($i$ = 1: $N$) do
        Generate a random number $rand_i$ in [0, 1]
        If ($rand_i < pu_i$) then
            Update the position of the $i$-th agent by Equations (1) and (4)
        Else
            Randomly select an agent $\boldsymbol{X}_{pt}$ from the top $N \times pt$ ($pt \in (0, 1)$) members in the population.
            Update the position of the $i$-th agent by Equations (5), (6) and (7)
        End if
    End for
    Evaluate the fitness of each agent by calculating the corresponding objective function value.
    Update $M_{pu}, M_F, M_{cr}$ using Equation (9)
    $k = k + 1$
    If ($k > H$) then
        $k = 1$
    End if
    Perform Levy flight disturbance for the current best position $\boldsymbol{X}_{cb}$ by Equation (11)
    Compute the opposite position $\bar{\boldsymbol{X}}_*$ for $\boldsymbol{X}_{cbLevy}$
    Update the current best position $\boldsymbol{X}_{cb}$
    Update $r_1, r_2, r_3, r_4$
    $t = t + 1$
End While
Return The global best position $\boldsymbol{X}_{gb}$
```

Results and Discussion

In this section, the performance of HGSCADE is evaluated on the CEC2014 benchmark functions. This test series contains 30 functions that can be categorized as unimodal functions (H1–H3), simple multimodal functions (H4–H16), hybrid functions (H17–H22), and composition functions (H23–H30) (Liang, Qu, and Suganthan 2013). The population size *N* is set as 50 and the maximum number of function evaluations is fixed as $10^4 \times$ D (D denotes the dimension of problem) for all the given problems. Fifty-one independent algorithm runs are executed for each problem. Moreover, the best value (the difference between the obtained best solution and the true optimal solution) for each run will be truncated to 0 if it is smaller than 10^{-8}. The statistical measures as mean and standard deviation (Std.) are reported, and the best is highlighted in bold.

Compared Algorithms

To verify the effectiveness and competitiveness of the proposed algorithm, it is compared with the standard SCA, other state-of-the-art MAs, and the variants of SCA:

- SCA: a population-based optimization algorithm which updates the population by sine and cosine functions (Mirjalili 2016b).
- DE: a powerful and straightforward evolutionary algorithm which was primarily designed for real-parameter optimization problems (Storn and Price 1997).
- ABC: a swarm-based optimization algorithm which simulates the intelligent foraging behavior of a honeybee swarm (Karaboga and Basturk 2007).
- GWO: a swarm intelligence algorithm which mimics the social behavior of gray wolves in chasing and hunting the prey (Mirjalili, Mirjalili, and Lewis 2014).
- WOA: a meta-heuristic algorithm which is inspired by the bubble-net hunting strategy of humpback whales (Mirjalili and Lewis 2016).
- SSA: a bio-heuristic algorithm which is inspired by the swarming behavior of salps when navigating and foraging in oceans (Mirjalili et al. 2017).
- HSCAPSO: a hybrid algorithm which adds memory-enabled behavior of PSO to SCA approach, and hybridizes SCA with PSO in iteration level (Nenavath, Kumar Jatoth, and Das 2018).
- HSCADE: a hybrid algorithm which incorporates DE operators into SCA in iteration level, and performs a greedy search (Nenavath and Jatoth 2018).
- PSOSCALF: a hybrid algorithm which combines PSO with the position updating equations of SCA and Levy flight (Rizk-Allah 2018).
- m-SCA: a hybrid algorithm which adds a self-adaptive component to SCA, and incorporates opposite perturbation into SCA (Gupta and Deep 2019).

The parameters settings for existing algorithms are extracted from the original reference. The parameter settings for the proposed HGSCADE algorithm are the combination of suggested parameter settings from SCA (a, r_2, r_3, r_4)

(Mirjalili 2016), SHADE (M_{pu}, M_F, M_{cr}, pt, H) (Tanabe and Fukunaga 2013), and PSOLF (β, $scale$) (Jensi and Jiji 2016). The parameter values for all algorithms are set in Table 2.

Test Results

The CEC2014 benchmark functions are considered with the dimension of D = 30 and D = 50. Due to space constraints, only the test results for D = 30 are reported in Table 3.

As seen in Table 3, HGSCADE provides the best results in terms of the mean for all unimodal functions (H1–H3). By introducing a greedy search, DE, and Levy flight-based local search, the population can exploit the nearby area of every individual best position. Therefore, the exploitation ability of HGSCADE is enhanced.

For simple multimodal functions, HGSCADE provides the best results in terms of mean on functions H4–H6, H8–H12, H15, and H16, ABC on functions H8, H13, and H14, and DE on functions H7 and H8. By inheriting the randomness of

Table 2. The parameters of algorithms and their values.

Algorithms	Parameters	Values
SCA	r_2, r_3, a, r_4	2π*rand, 2*rand, 2, rand
DE	F, cr	[0.2, 0.8], 0.2
ABC	φ, limit	[−1,1], 20
GWO	r_1, r_2, a	rand, rand, 2 − 2*t/T
WOA	a, a_2, l, b, p	2 − 2* t/T, −1 − t/T, (a_2 − 1)*rand + 1, 1, rand
SSA	c_2, c_3	rand, rand
HSCAPSO	r_2, r_3, a, r_4, ω, c_1, c_2, v_{max}	2π*rand, 2*rand, 2, rand, 0.9 to 0.4, 2, 2, 4
HSCADE	r_2, r_3, a, r_4, F, cr	2π*rand, 2*rand, 2, rand, [0.2, 0.8], 0.2
PSOSCALF	r_2, r_3, a, r_4, ω_{max}, ω_{min}	2π*rand, 2*rand, 2, rand, 0.9, 0.4
	c_{1max}, c_{1min}, c_{2max}, c_{2min},	0.5, 2.5, 0.5, 2.5
	K, limit, β, scale	10,10, 1.5, 0.01
m-SCA	r_2, r_3, a, r_4, S_R, J_R	2π*rand, 2*rand, 2, rand, rand, 0.1
HGSCADE	r_2, r_3, a, r_4	2π*rand, 2*rand, 2, rand,
	M_{pu}, M_F, M_{cr}, pt, H, β, scale	0.5, 0.5, 0.5, 20, 10, 1.5, 0.01

Table 3. Wilcoxon signed rank test between HGSCADE and other algorithms for the CEC2014 functions with D = 30 at a 5% level of significance.

	Cylindricity error (mm)				
Methods	Data set 1	Data set 2	Data set 3	Data set 4	Data set 5
SCA	0.2032	0.5754	0.0330	0.0168	0.8600
DE	0.0083	0.0196	0.0320	0.0164	0.0158
ABC	0.0239	0.1385	0.1376	0.1570	0.1432
GWO	0.0032	0.1252	0.0321	0.0164	0.2288
WOA	0.1847	0.4239	0.0350	0.0179	1.8552
SSA	0.002788	0.0195	0.03184	0.0164	0.0154
HSCAPSO	10.5857	9.4527	21.9388	15.1798	21.0804
HSCADE	0.0968	0.4573	0.0325	0.0166	0.6378
PSOSCALF	0.002788	0.0196	0.0320	0.0164	0.0146
m-SCA	0.0671	0.1167	0.0321	0.0165	0.3051
HGSCADE	0.002788	0.0194	0.03183	0.0164	0.0143
LSM	0.037162	0.0234	0.0366	0.0195	0.0168
Published results	0.002788	0.01938	0.0319	0.0167	0.0143

SCA and Levy flight disturbance, the population can explore the whole search space. Therefore, the exploration ability of HGSCADE is enhanced.

HGSCADE provides the best results in terms of the mean for all hybrid functions (H17–H22). For composition functions, HGSCADE provides the best results in terms of mean on functions H23, H25, H26, H29, and H30, DE on functions H23, H26–28, ABC on functions H23 and H26, and SSA on function H26. For function H24, SCA, GWO, and m-SCA give the same best results in terms of the mean. By the success history-based adaptation of parameter pu_i, each agent is automatically updated by the SCA method or DE method to balance the trade-off between exploration and exploitation.

In addition, it can be observed that HGSCADE is the best algorithm in terms of overall standard deviation, i.e., HGSCADE has the highest stability. The opposition-based population initialization, greedy search, and success history-based parameter adaptation improve the stability of HGSCADE.

Statistical Analysis

Considering the effect of each run, a non-parametric statistical test called Wilcoxon signed rank test is implemented at a significance level of 5%. The calculated p-value determines to accept or reject the null hypothesis. The grade symbols +, ≈ and – signify that HGSCADE is superior, similar, and inferior to a compared algorithm, respectively. The character string N/A signifies that the two samples are identical and the Wilcoxon signed rank test is invalid. Accordingly, the Wilcoxon signed rank test results are presented in Table 4. The grades are summarized in the last row of the table.

From Table 4, HGSCADE performs better than HSCAPSO, HSCADE, and PSOSCALF in all the CEC2014 benchmark functions for D = 30. In 29, 22, 26, 29, 28, 26, and 28 out of 30 CEC2014 benchmark functions for D = 30, HGSCADE performs better than SCA, DE, ABC, GWO, WOA, SSA, and m-SCA, respectively. It verifies the test results that HGSCADE is the best algorithm in comparison with other algorithms.

Convergence Analysis

Considering that there are too many functions, four functions (H3, H13, H20, and H30) from different categories are selected to exemplify the convergence behavior. The median of the obtained best values is displayed on the vertical axis in logarithmic scale, while the number of function evaluations is displayed on the horizontal axis. The convergence curves for the selected functions are shown in Figures 1–2.

It can be observed that HGSCADE has a faster convergence speed to produce an accurate solution than that of compared algorithms except for H13. It indicates that HGSCADE has the ability to create a balance between exploration and exploitation. The opposition-based population initialization provides initial solutions with favorable distribution, and the greedy search prevents excluding good

Table 4. The cylindricity error evaluation results of different algorithms.

No.	Criterion	SCA	DE	ABC	GWO	WOA	SSA	HSCAPSO	HSCADE	PSOSCALF	m-SCA	HGSCADE
H1	Mean	2.62E+08	1.77E+07	1.12E+07	7.57E+07	3.15E+07	1.62E+06	3.84E+07	3.89E+08	1.99E+07	2.04E+07	**8.63E+03**
	Std.	7.37E+07	3.91E+06	3.17E+06	5.26E+07	1.42E+07	7.29E+05	2.95E+07	6.96E+07	2.19E+07	6.87E+06	**7.17E+03**
H2	Mean	1.81E+10	3.36E-01	2.19E+02	4.23E+09	4.69E+06	8.71E+03	1.06E+07	2.72E+10	1.84E+05	2.71E+08	**0.00E+00**
	Std.	3.39E+09	2.29E+00	2.25E+02	3.14E+09	8.04E+06	7.61E+03	3.53E+06	3.70E+09	1.06E+06	5.34E+07	**0.00E+00**
H3	Mean	4.15E+04	1.89E+02	6.50E+02	2.93E+04	4.06E+04	2.11E+03	5.33E+04	7.57E+04	1.76E+04	2.50E+04	**0.00E+00**
	Std.	8.35E+03	1.89E+02	4.83E+02	9.80E+03	2.78E+04	1.07E+03	1.29E+04	1.22E+04	1.29E+04	5.64E+03	**0.00E+00**
H4	Mean	1.16E+03	1.14E+02	1.60E+01	2.90E+02	1.86E+02	7.97E+01	2.72E+02	2.87E+03	1.51E+02	1.72E+02	**9.03E+00**
	Std.	2.43E+02	1.68E+01	2.11E+01	1.42E+02	6.06E+01	2.96E+01	1.23E+02	4.84E+02	4.96E+01	**1.54E+01**	2.39E+01
H5	Mean	2.09E+01	2.06E+01	2.04E+01	2.09E+01	2.03E+01	**2.01E+01**	2.11E+01	2.09E+01	2.09E+01	2.09E+01	**2.01E+01**
	Std.	5.70E-02	4.51E-02	4.35E-02	4.49E-02	2.00E-01	1.04E-01	9.05E-02	5.85E-02	4.99E-02	5.80E-02	**3.03E-02**
H6	Mean	3.48E+01	1.84E+01	1.51E+01	1.39E+01	3.53E+01	2.02E+01	2.48E+01	3.65E+01	2.52E+01	1.86E+01	**1.13E+01**
	Std.	2.85E+00	**1.28E+00**	1.53E+00	3.40E+00	3.83E+00	3.77E+00	3.91E+00	1.44E+00	2.93E+00	2.34E+00	2.25E+00
H7	Mean	1.48E+02	**6.05E-09**	4.80E-06	4.11E+01	1.01E+00	1.12E-02	1.10E+00	2.49E+02	5.55E-01	4.06E+00	1.35E-03
	Std.	2.73E+01	**4.18E-08**	1.02E-05	3.59E+01	5.22E-02	9.37E-03	3.50E-02	3.26E+01	1.52E+00	7.34E-01	4.33E-03
H8	Mean	2.45E+02	**0.00E+00**	**0.00E+00**	8.10E+01	1.81E+02	1.23E+02	7.99E+01	2.75E+02	9.12E+01	9.85E+01	**0.00E+00**
	Std.	1.53E+01	**0.00E+00**	**0.00E+00**	1.98E+01	3.44E+01	2.98E+01	1.63E+01	1.41E+01	2.09E+01	1.29E+01	**0.00E+00**
H9	Mean	2.77E+02	1.14E+02	8.41E+01	1.04E+02	2.44E+02	1.31E+02	1.03E+02	3.19E+02	1.33E+02	1.23E+02	**3.55E+01**
	Std.	2.12E+01	1.02E+01	1.04E+01	2.24E+01	6.53E+01	3.15E+01	2.81E+01	1.34E+01	3.15E+01	1.38E+01	**6.75E+00**
H10	Mean	6.17E+03	5.91E+01	1.34E+00	2.12E+03	3.97E+03	3.41E+03	3.55E+03	6.48E+03	3.19E+03	4.03E+03	**1.35E-02**
	Std.	4.86E+02	2.74E+01	8.02E-01	5.22E+02	5.88E+02	5.82E+02	7.19E+02	2.56E+02	9.51E+02	4.87E+02	**1.81E-02**
H11	Mean	7.02E+03	4.76E+03	2.17E+03	2.63E+03	4.66E+03	3.49E+03	5.77E+03	6.99E+03	6.59E+03	4.96E+03	**1.56E+03**
	Std.	2.97E+02	**2.36E+02**	2.64E+02	4.85E+02	9.02E+02	7.33E+02	1.31E+03	2.96E+02	7.64E+02	5.64E+02	3.71E+02
H12	Mean	2.41E+00	1.01E+00	4.24E-01	1.55E+00	1.62E+00	4.54E-01	2.93E+00	2.40E+00	2.54E+00	1.85E+00	**1.71E-01**
	Std.	2.99E-01	1.35E-01	5.60E-02	1.13E+00	5.14E-01	2.17E-01	7.54E-01	2.65E-01	2.97E-01	2.72E-01	**2.95E-02**
H13	Mean	3.10E+00	3.50E-01	**2.47E-01**	5.13E-01	4.97E-01	4.73E-01	5.25E-01	4.06E+00	7.25E-01	4.62E-01	2.55E-01
	Std.	3.25E-01	3.55E-02	**2.50E-02**	3.78E-01	1.05E-01	9.95E-02	1.05E-01	2.97E-01	1.70E-01	6.28E-02	5.87E-02
H14	Mean	4.85E+01	2.78E-01	**1.96E-01**	7.95E+00	2.77E-01	2.76E-01	3.25E-01	7.99E+01	7.62E-01	3.16E-01	2.68E-01
	Std.	9.40E+00	2.93E-02	**1.60E-02**	9.71E+00	8.98E-02	4.25E-02	6.23E-02	9.06E+00	3.55E-01	4.35E-02	4.33E-02
H15	Mean	4.23E+03	1.26E+01	1.09E+01	2.74E+02	7.13E+01	7.89E+00	2.34E+01	1.03E+05	4.42E+01	1.90E+01	**3.55E+00**
	Std.	4.53E+03	**7.89E-01**	1.27E+00	5.13E+02	2.34E+01	3.22E+00	2.54E+00	5.01E+04	1.68E+01	1.61E+00	9.77E-01
H16	Mean	1.29E+01	1.13E+01	1.07E+01	1.08E+01	1.26E+01	1.17E+01	1.27E+01	1.30E+01	1.28E+01	1.22E+01	**9.60E+00**
	Std.	2.80E-01	3.70E-01	2.59E-01	7.39E-01	4.74E-01	6.41E-01	4.69E-01	**1.59E-01**	3.88E-01	2.21E-01	6.60E-01
H17	Mean	6.40E+06	1.21E+06	2.83E+06	1.63E+06	3.44E+06	1.14E+05	1.61E+06	9.37E+06	2.57E+06	5.08E+05	**1.11E+03**
	Std.	2.68E+06	4.48E+05	1.33E+06	1.58E+06	2.29E+06	9.11E+04	9.68E+05	3.05E+06	2.11E+06	2.52E+05	**4.76E+02**
H18	Mean	1.93E+08	7.06E+03	6.98E+03	8.49E+06	4.80E+03	2.01E+03	5.35E+03	2.61E+08	6.82E+04	6.44E+05	**6.89E+01**

(*Continued*)

Table 4. (Continued).

No.	Criterion	SCA	DE	ABC	GWO	WOA	SSA	HSCAPSO	HSCADE	PSOSCALF	m-SCA	HGSCADE
	Std.	1.01E+08	4.55E+03	4.10E+03	1.95E+07	4.86E+03	2.21E+03	5.39E+03	8.68E+07	2.50E+05	3.80E+05	**3.25E+01**
H19	Mean	8.90E+01	7.92E+00	7.39E+00	4.10E+01	4.04E+01	1.53E+01	6.06E+01	1.32E+02	2.77E+01	1.51E+01	**4.23E+00**
	Std.	2.26E+01	**6.36E-01**	6.56E-01	2.44E+01	3.07E+01	9.89E+00	3.25E+01	2.37E+01	2.21E+01	1.64E+00	1.19E+00
H20	Mean	1.62E+04	2.49E+03	4.87E+03	1.63E+04	2.25E+04	3.86E+02	2.52E+04	2.52E+04	1.55E+04	6.91E+03	**9.80E+01**
	Std.	5.39E+03	1.10E+03	1.92E+03	9.54E+03	1.48E+04	1.17E+02	1.37E+04	9.19E+03	7.45E+03	2.82E+03	**5.27E+02**
H21	Mean	1.74E+06	1.91E+05	4.37E+05	6.05E+05	1.03E+06	6.66E+04	4.58E+05	2.08E+06	7.84E+05	9.11E+04	**3.51E+02**
	Std.	9.87E+05	9.69E+04	1.56E+05	1.00E+06	8.70E+05	4.66E+04	2.90E+05	1.01E+06	7.57E+05	4.47E+04	**1.85E+02**
H22	Mean	8.45E+02	1.48E+02	3.03E+02	3.75E+02	8.02E+02	3.87E+02	5.40E+02	8.31E+02	5.10E+02	1.79E+02	**1.43E+02**
	Std.	1.65E+02	6.84E+01	8.43E+01	1.57E+02	2.29E+02	1.44E+02	1.92E+02	1.25E+02	1.95E+02	**6.30E+01**	8.24E+01
H23	Mean	3.75E+02	**3.15E+02**	**3.15E+02**	3.38E+02	3.31E+02	3.15E+02	3.34E+02	4.49E+02	3.16E+02	3.22E+02	**3.15E+02**
	Std.	1.45E+01	**4.02E-13**	1.42E-01	1.34E+01	7.52E+00	5.07E-02	1.53E+01	2.65E+01	2.33E+00	9.43E-01	8.10E-13
H24	Mean	**2.00E+02**	2.24E+02	2.27E+02	**2.00E+02**	2.06E+02	2.33E+02	2.37E+02	3.25E+02	2.50E+02	**2.00E+02**	2.27E+02
	Std.	3.11E-01	8.74E-01	2.58E+00	**8.64E-04**	3.93E+00	9.02E+00	1.12E+01	1.09E+01	1.01E+01	5.46E-03	3.61E+00
H25	Mean	2.30E+02	2.08E+02	2.09E+02	2.10E+02	2.19E+02	2.13E+02	2.21E+02	2.44E+02	2.16E+02	2.11E+02	**2.06E+02**
	Std.	6.25E+00	1.01E+00	**9.05E-01**	5.19E+00	1.59E+01	2.70E+00	5.99E+00	6.40E+00	4.80E+00	1.71E+00	2.73E+00
H26	Mean	1.03E+02	**1.00E+02**	**1.00E+02**	1.43E+02	1.08E+02	**1.00E+02**	1.01E+02	1.04E+02	1.01E+02	1.01E+02	**1.00E+02**
	Std.	6.32E-01	**3.72E-02**	6.09E-02	4.97E+01	4.38E+01	1.07E-01	1.37E-01	2.81E-01	1.62E-01	5.31E-02	5.83E-02
H27	Mean	7.84E+02	**3.77E+02**	4.04E+02	6.67E+02	1.03E+03	7.04E+02	7.20E+02	7.63E+02	7.68E+02	4.20E+02	4.23E+02
	Std.	3.23E+02	6.36E+01	**5.55E+01**	1.34E+02	3.82E+02	1.80E+02	2.47E+02	1.47E+02	3.04E+02	5.97E+00	5.96E+01
H28	Mean	1.92E+03	**8.41E+02**	9.64E+02	1.23E+03	2.10E+03	1.10E+03	3.11E+03	2.51E+03	1.57E+03	1.09E+03	8.48E+02
	Std.	3.27E+02	**2.80E+01**	4.86E+01	3.01E+02	5.05E+02	2.93E+02	9.10E+02	3.70E+02	4.12E+02	4.13E+01	3.46E+01
H29	Mean	1.04E+07	1.76E+03	1.14E+03	1.62E+06	6.27E+06	3.15E+06	3.83E+07	1.71E+07	4.91E+06	3.59E+04	**9.68E+02**
	Std.	6.27E+06	1.90E+02	**1.14E+02**	3.68E+06	4.73E+06	5.86E+06	6.97E+07	5.59E+06	7.37E+06	1.56E+04	2.96E+02
H30	Mean	2.45E+05	2.74E+03	3.96E+03	5.81E+04	9.00E+04	9.62E+03	7.05E+04	3.27E+05	3.67E+04	2.06E+04	**1.86E+03**
	Std.	8.35E+04	**7.18E+02**	9.68E+02	4.93E+04	6.94E+04	1.03E+04	8.23E+04	7.94E+04	3.42E+04	6.36E+03	8.94E+02

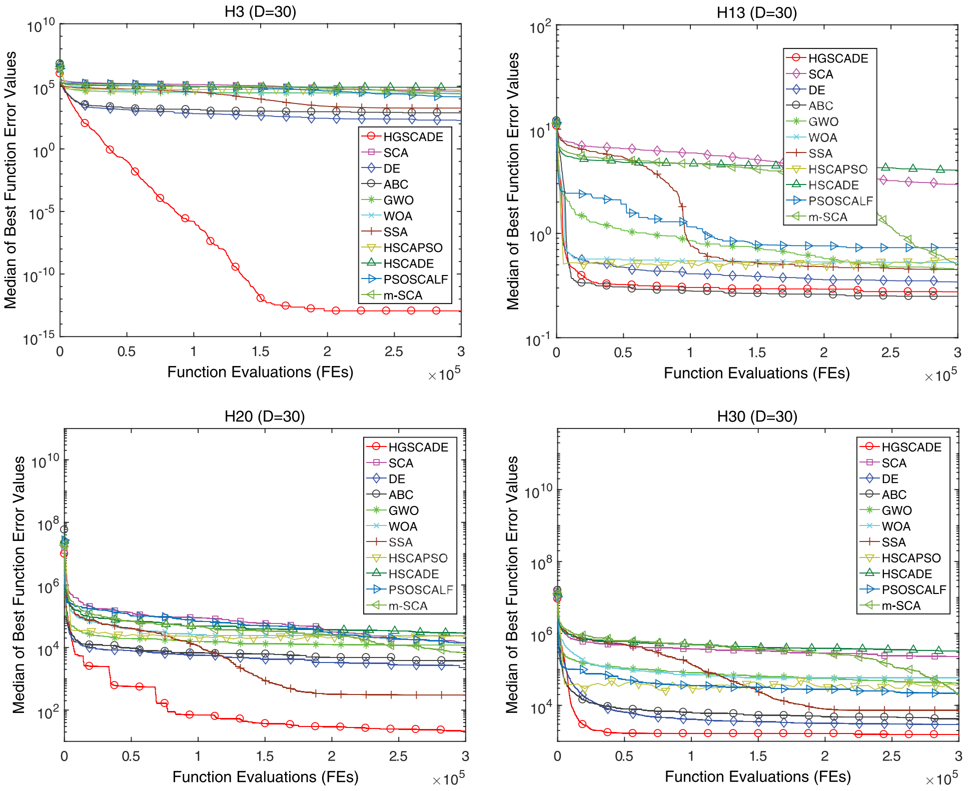

Figure 1. The convergence curves of all algorithms for certain CEC2014 functions (H3, H13, H20, and H30) with D = 30.

solutions; thus, the convergence speed is accelerated. The DE enhances the exploitation ability and maintains the population diversity, the Levy flight-based local search avoids local optima, and the success history-based parameter adaptation strengthens the stability; thus, the solution accuracy is improved.

Complexity Analysis

The complexity of HGSCADE (represented by notation O) can be calculated as O (HGSCADE) = O (positions initializing by OBL) + O (positions updating by greedy SCA) + O (positions updating by DE) + O (global best position updating) + O (current best position updating by local search), i.e.

$$O(\text{HGSCADE}) = O[N + T \times (N \times D + 2N + 2)] \tag{15}$$

where N is the population size, D is the dimension of the problem and T is the maximum number of iterations. Similarly, the complexity of SCA is $O(\text{SCA}) = O[T \times (N \times D + N)]$. The opposition-based population initialization, the greedy search, and the Levy flight-based local search increase the computational complexity of the proposed algorithm.

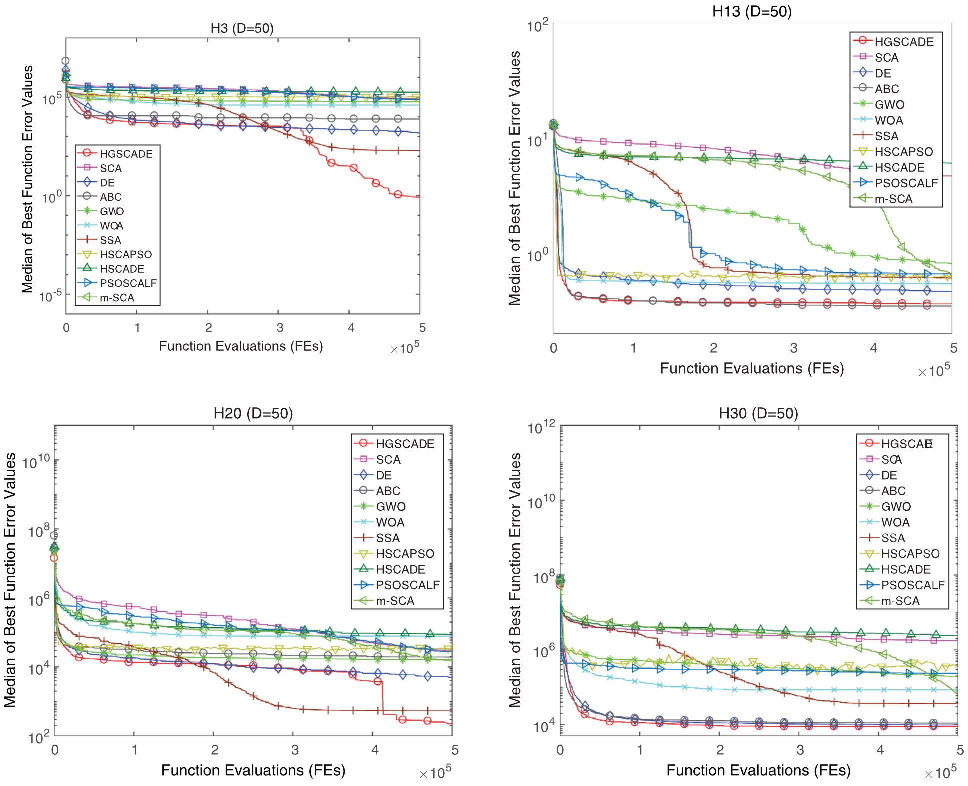

Figure 2. The convergence curves of all algorithms for certain CEC2014 functions (H3, H13, H20, and H30) with D = 50.

HGSCADE for Cylindricity Error Evaluation

Mathematical Model of Cylindricity Error

As shown in Figure 3, the cylindricity error f is defined as the minimum radius difference between two coaxial theoretical cylinders which envelop the actual contour of a cylindrical part. It is assumed that the measured points in the Cartesian coordinate system are $p_i(x_i, y_i, z_i)$ $(i = 1, 2, \ldots, n)$, where x_i, y_i, z_i are coordinate values, and n is the number of measured points. Supposing that the fixed point of L is (x_0, y_0, z_0) and the directional vector of L is $[a, b, c]$. Then, the parametric equation of L can be defined as:

$$\frac{x - x_0}{a} = \frac{y - y_0}{b} = \frac{z - z_0}{c} \tag{16}$$

The distance from point p_i to axis L can be calculated by:

$$r = \sqrt{(A^2 + B^2 + C^2)/(a^2 + b^2 + c^2)} \tag{17}$$

where $A = (y_i - y_0) \times c - (z_i - z_0) \times b$, $B = (z_i - z_0) \times a - (x_i - x_0) \times c$, $C = (x_i - x_0) \times b - (y_i - y_0) \times a$.

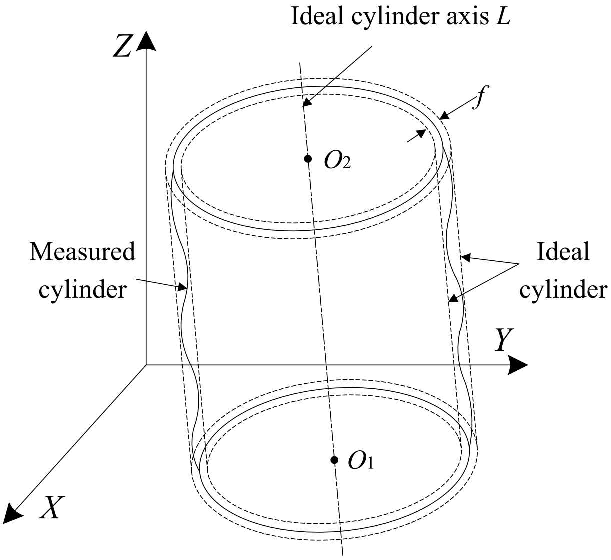

Figure 3. Schematic of cylindricity error.

Finally, the cylindricity error is defined as:

$$f(x_0, y_0, z_0, a, b, c) = \min(\max(r) - \min(r)) \tag{18}$$

Data Sets

Five original-measured data sets from corresponding literature are used to experiment. These data sets are named as Data set 1 (24 samples cited from Lai et al. 2000), Data set 2 (32 samples cited from Yang et al. 2018), Data set 3 (80 samples cited from Lei et al. 2011), Data set 4 (24 samples cited from; Li et al. 2009), and Data set 5 (60 samples cited from Zhao, Wen, and Xu 2015), respectively.

Experiments and Analysis

HGSCADE is used for cylindricity error evaluation and compared with other algorithms. For all algorithms, the population size is fixed to 30, the search space of variables (x_0, y_0, z_0) is set as the data range, and the search space of variables (a, b, c) is set from – 1 to 1. The search will be terminated when the number of evaluations reaches 10000. The evaluation results (the mean of 30 independent evaluation values for each data set) of HGSCADE and other comparison algorithms are listed in Table 5. The convergence curves of HGSCADE and other comparison algorithms are shown in Figure 4.

As can be observed from Table 5, the cylindricity error evaluation results of HGSCADE are much better than that of SCA, ABC, WOA, HSCAPSO, HSCADE,

Table 5. Mean and standard deviation of 51 independent runs for all algorithms on the CEC2014 functions with D = 30.

		HGSCADE versus								
No.		SCA	DE	ABC	GWO	WOA	SSA	HSCAPSO	HSCADE	PSOSCALF
H1	*p*-Values	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	+	+	+	+	+	+	+	+
H2	*p*-Values	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	+	+	+	+	+	+	+	+
H3	*p*-Values	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	+	+	+	+	+	+	+	+
H4	*p*-Values	0.000000	0.000000	0.000512	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	+	+	+	+	+	+	+	+
H5	*p*-Values	0.000000	0.000000	0.000000	0.000000	0.000000	0.000026	0.000000	0.000000	0.000000
	Grades	+	+	+	+	+	-	+	+	+
H6	*p*-Values	0.000000	0.000000	0.000000	0.000389	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	+	+	+	+	+	+	+	+
H7	*p*-Values	0.000000	0.078125	0.000140	0.000000	0.000000	0.000026	0.000000	0.000000	0.000000
	Grades	+	≈	+	+	+	-	+	+	+
H8	*p*-Values	0.000000	N/A	N/A	0.000000	0.000000	0.000026	0.000000	0.000000	0.000000
	Grades	+	≈	≈	+	+	-	+	+	+
H9	*p*-Values	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	+	+	+	+	+	+	+	+
H10	*p*-Values	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	+	+	+	+	+	+	+	+
H11	*p*-Values	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	+	+	+	+	+	+	+	+
H12	*p*-Values	0.000000	0.000000	0.000000	0.000001	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	+	+	+	+	+	+	+	+
H13	*p*-Values	0.000000	0.000000	0.404705	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	+	≈	+	+	+	+	+	+
H14	*p*-Values	0.000000	0.277409	0.000000	0.000000	0.679119	0.349193	0.000009	0.000000	0.000000
	Grades	+	≈	-	+	≈	≈	+	+	+
H15	*p*-Values	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	+	+	+	+	+	+	+	+
H16	*p*-Values	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	+	+	+	+	+	+	+	+
H17	*p*-Values	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	+	+	+	+	+	+	+	+

(*Continued*)

Table 5. (Continued).

		HGSCADE versus								
No.		SCA	DE	ABC	GWO	WOA	SSA	HSCAPSO	HSCADE	PSOSCALF
H18	*p*-Values	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	+	+	+	+	+	+	+	+
H19	*p*-Values	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	+	+	+	+	+	+	+	+
H20	*p*-Values	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	+	+	+	+	+	+	+	+
H21	*p*-Values	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	+	+	+	+	+	+	+	+
H22	*p*-Values	0.000000	0.856213	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	≈	+	+	+	+	+	+	+
H23	*p*-Values	0.000000	0.003906	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	-	+	+	+	+	+	+	+
H24	*p*-Values	0.000000	0.000000	0.012990	0.000000	0.000000	0.000018	0.000000	0.000000	0.000000
	Grades	-	-	+	-	≈	+	+	+	+
H25	*p*-Values	0.000000	0.000027	0.000001	0.000001	0.000002	0.000000	0.000000	0.000000	0.000000
	Grades	+	+	+	+	+	+	+	+	+
H26	*p*-Values	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	+	+	+	+	+	+	+	+
H27	*p*-Values	0.000000	0.000645	0.363840	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	-	≈	+	+	+	+	+	+
H28	*p*-Values	0.000000	0.415328	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	≈	+	+	+	+	+	+	+
H29	*p*-Values	0.000000	0.000000	0.001009	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	+	+	+	+	+	+	+	+
H30	*p*-Values	0.000000	0.000006	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Grades	+	+	+	+	+	+	+	+	+
Grades + / ≈ / -		29/0/1	22/5/3	26/3/1	29/0/1	28/1/1	26/1/3	30/0/0	30/0/0	30/0/0

and LSC for all the data sets. Although SSA and PSOSCALF provide the same results as HGSCADE for Data set 1 and Data set 4, DE and GWO provide the same results as HGSCADE for Data set 4, they cannot give the best results for all the data sets. Moreover, the results of HGSCADE are well in agreement with the published results (equally for Data set 1, Data set 4 and Data set 5, slightly worse for Data set 2, slightly better for Data set 3). It can be observed from Figure 4 that the evaluation speed of HGSCADE is not the fastest, but HGSCADE can give the best evaluation results with relative efficiency. HGSCADE keeps a very good balance between exploration and exploitation in cylindricity error evaluation, so satisfactory results

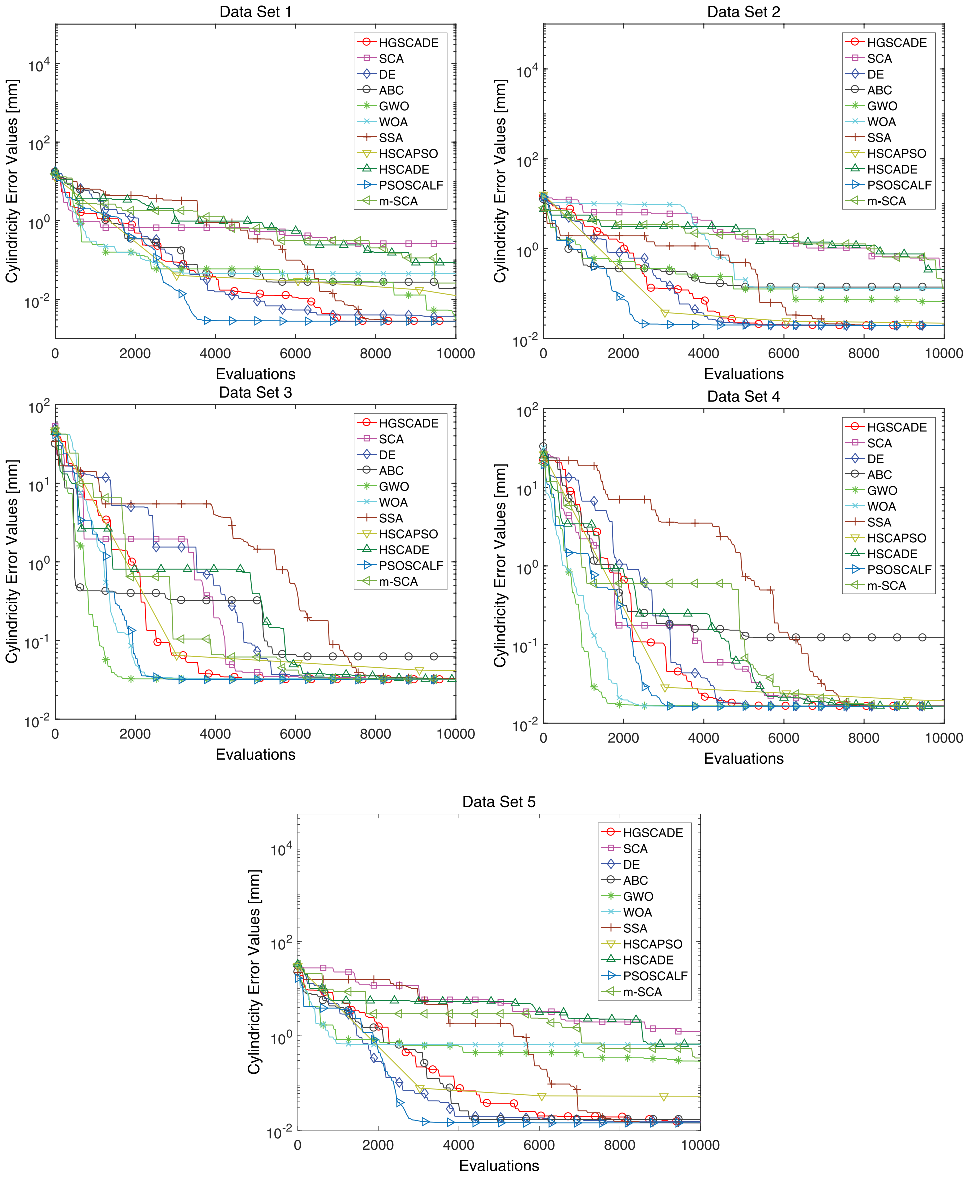

Figure 4. The convergence curves of all algorithms for cylindricity error evaluation.

are obtained. The availability of HGSCADE for cylindricity error evaluation is verified by the experiment.

Conclusions

In this study, a novel HGSCADE algorithm is proposed for solving optimization problems and evaluating cylindricity error. The proposed algorithm combines SCA with five strategies: (1) Opposition-based learning (2) Greedy search (3) Differential evolution (4) Success history-based parameter adaptation (5) Local search based on Levy flight. HGSCADE is compared with the standard SCA, other state-of-the-art MAs, and the variants of SCA on the CEC2014 benchmark functions. The comparison and analysis of the test results prove that the performance of HGSCADE is better than that of other compared algorithms. In addition, the proposed algorithm is applied for cylindricity error evaluation of five measured cylindrical parts and is verified to be an effective mean to evaluate cylindricity error.

However, the test results of HGSCADE for several CEC2014 functions are worse than that of certain-compared algorithms (SSA for H5, H7, and H8; DE for H23, H24, and H27; ABC for H14; SCA, GWO, WOA, and m-SCA for H24). This suggests that HGSCADE accords with the NFL theorem. The future work will be concentrated in two directions: (1) Attempting to apply HGSCADE in solving other engineering optimization problems (2) Extending the study of SCA for solving more complex problems.

Disclosure Statement

The authors declare that they have no conflict of interest.

Funding

This work was supported by the National Natural Science Foundation of China under Grant [51565033]; and the Natural Science Foundation of Gansu Province under Grant [18JR3RA142].

ORCID

Qijun Li http://orcid.org/0000-0001-7836-5462
Baolin Dai http://orcid.org/0000-0003-4332-135X

References

Abd Elaziz, M., D. Oliva, and S. W. Xiong. 2017. An improved opposition-based sine cosine algorithm for global optimization. *Expert Systems with Applications* 90:484–500. doi:10.1016/j.eswa.2017.07.043.

Attia, A. F., R. A. El Sehiemy, and H. M. Hasanien. 2018. Optimal power flow solution in power systems using a novel sine-cosine algorithm. *International Journal of Electrical Power & Energy Systems* 99:331–43.

Chegini, S. N., A. Bagheri, and F. Najafi. 2018. PSOSCALF: A new hybrid PSO based on sine cosine algorithm and Levy flight for solving optimization problems. *Applied Soft Computing* 73:697–726.

Das, S., A. Bhattacharya, and A. K. Chakraborty. 2018. Solution of short-term hydrothermal scheduling using sine cosine algorithm. *Soft Computing* 22:6409–27.

Dorigo, M., and G. Di Caro. 1999. Ant colony optimization: A new meta-heuristic. *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, Washington, DC, USA, 1470–77. IEEE.

Geem, Z. W., J. H. Kim, and G. V. Loganathan. 2001. A new heuristic optimization algorithm: Harmony search. *Simulation* 76 (2):60–68.

Gupta, S., and K. Deep. 2019. A hybrid self-adaptive sine cosine algorithm with opposition based learning. *Expert Systems with Applications* 119:210–30.

Issa, M., A. E. Hassanien, D. Oliva, A. Helmi, I. Ziedan, and A. Alzohairy. 2018. ASCA-PSO: Adaptive sine cosine optimization algorithm integrated with particle swarm for pairwise local sequence alignment. *Expert Systems with Applications* 99:56–70.

Jensi, R., and G. W. Jiji. 2016. An enhanced particle swarm optimization with levy flight for global optimization. *Applied Soft Computing* 43:248–61.

Karaboga, D., and B. Basturk. 2007. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization* 39:459–71.

Kennedy, J., and R. Eberhart. 1995. Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, WA, Australia, 1942–48. IEEE.

Lai, H. Y., W. Y. Jywe, C. K. Chen, and C. H. Liu. 2000. Precision modeling of form errors for cylindricity evaluation using genetic algorithms. *Precision Engineering* 24 (4):310–19.

Lei, X. Q., H. W. Song, Y. J. Xue, J. S. Li, J. Zhou, and M. D. Duan. 2011. Method for cylindricity error evaluation using geometry optimization searching algorithm. *Measurement* 44 (9):1556–63.

Li, J. S., X. Q. Lei, Y. J. Xue, and M. D. Duan. 2009. Evaluation algorithm of cylindricity error based on coordinate transformation. *China Mechanical Engineering* 20 (16):1983–87.

Li, S., H. J. Fang, and X. Y. Liu. 2018. Parameter optimization of support vector regression based on sine cosine algorithm. *Expert Systems with Applications* 91:63–77.

Liang, J. J., B. Y. Qu, and P. N. Suganthan. 2013. *Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization*. Singapore: Zhengzhou University, Zhengzhou, China and Nanyang Technological University.

Mirjalili, S. 2015. The ant lion optimizer. *Advances in Engineering Software* 83:80–98.

Mirjalili, S. 2016. SCA: A sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems* 96:120–33.

Mirjalili, S., A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili. 2017. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software* 114:163–91.

Mirjalili, S., and A. Lewis. 2016. The whale optimization algorithm. *Advances in Engineering Software* 95:51–67.

Mirjalili, S., S. M. Mirjalili, and A. Lewis. 2014. Grey wolf optimizer. *Advances in Engineering Software* 69:46–61.

Nenavath, H., and R. K. Jatoth. 2018. Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking. *Applied Soft Computing* 62:1019–43.

Nenavath, H., D. R. Kumar Jatoth, and D. S. Das. 2018. A synergy of the sine-cosine algorithm and particle swarm optimizer for improved global optimization and object tracking. *Swarm and Evolutionary Computation* 43:1–30.

Passino, K. M. 2002. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine* 22 (3):52–67.

Rashedi, E., H. Nezamabadi-pour, and S. Saryazdi. 2009. GSA: A gravitational search algorithm. *Information Sciences* 179 (13):2232–48.

Rizk-Allah, R. M. 2018. Hybridizing sine cosine algorithm with multi-orthogonal search strategy for engineering design problems. *Journal of Computational Design and Engineering* 5 (2):249–73.

Simon, D. 2008. Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation* 12 (6):702–13.

Sindhu, R., R. Ngadiran, Y. M. Yacob, N. A. H. Zahri, and M. Hariharan. 2017. Sine-cosine algorithm for feature selection with elitism strategy and new updating mechanism. *Neural Computing and Applications* 28:2947–58.

Srinivas, M., and L. M. Patnaik. 1994. Genetic algorithms: A survey. *Computer* 27 (6):17–26.

Storn, R., and K. Price. 1997. Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11:341–59.

Tanabe, R., and A. Fukunaga. 2013. Success-history based parameter adaptation for differential evolution. *Proceedings of 2013 IEEE Congress on Evolutionary Computation*, Cancún, México, 71–78. IEEE.

Tizhoosh, H. R. 2005. Opposition-based learning: A new scheme for machine intelligence. *Proceedings of International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce* (*CIMCA-IAWTIC'06*), Vienna, Austria, 695–701. IEEE.

Wen, X. L., J. C. Huang, D. H. Sheng, and F. L. Wang. 2010. Conicity and cylindricity error evaluation using particle swarm optimization. *Precision Engineering* 34 (2):338–44.

Wolpert, D. H., and W. G. Macready. 1997. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1 (1):67–82.

Yang, X. S. 2008. *Introduction to mathematical optimization: From linear programming to metaheuristics*. Cambridge: Cambridge International Science Publishing.

Yang, Y., M. Li, C. Wang, and Q. Y. Wei. 2018. Cylindricity error evaluation based on an improved harmony search algorithm. *Scientific Programming* 2:1–13.

Zhao, Y. B., X. L. Wen, and Y. X. Xu. 2015. Cylindricity error inspection and evaluation based on CMM and QPA. *China Mechanical Engineering* 26 (18):2432–36.