# Elite and dynamic opposite learning enhanced sine cosine algorithm for application to plat-fin heat exchangers design problem

Lidong Zhang[1] · Tianyu Hu[1] · Zhile Yang[2] ● · Dongsheng Yang[3] · Jianhua Zhang[4]

## Abstract

The heat exchanger has been widely used in the energy and chemical industry and plays an irreplaceable role in the featured applications. The design of heat exchanger is a mixed integer complex optimization problem, where the efficient design significantly improves the efficiency and reduces the cost. Many intelligent methods have been developed for heat exchanger optimal design. In this paper, a novel variant of sine and cosine algorithm named EDOLSCA is proposed, enhanced by dynamic opposite learning algorithm and the elite strategy. The proposed method is tested in CEC2014 benchmark and proved to be of significant advantages over the original algorithm. The new algorithm is then validated in the plate-fin heat exchanger (PFHE) optimal design problem. The comparison results of the proposed algorithm and other algorithms prove that EDOLSCA also has demonstrated superiority in heat exchanger optimal design.

**Keywords** Plate-fin heat exchanger · Design optimization · Sine cosine algorithm · Dynamic-opposite learning

## 1 Introduction

In 2020s, energy consumption has been growing at an exponential rate thanks to the higher energy efficiency and strong demand. As early as 1973, Lincoln et al. [28] pointed out that energy conservation should be paid attention to and the sustainable development of environment and resources should be guaranteed through technological progress and social science [55]. With the significant improvement of computational capacity, an increasing number of industrial equipment began to use computer-aided design and optimization to improve the energy efficiency and reduce economic expenditure [49, 51]. Plate-fin heat exchanger (PFHE) is a common type of heat exchanger, also known as brazed aluminum heat exchanger in the thermal engineering. Compared with other types of heat exchangers, PFHE has compact structure, relatively small size, relatively high thermal load and lightweight, due to which it has been widely used in aerospace, railway, energy and chemical industries. In real-world cases, the majority of the design for heat exchanger is based on the experience of designers, without any unified design method and scheme [45]. Bio-optimization, which is of high compatible capacity in solving various optimization problems, is promising to effectively solve PFHE optimal design problem. To solve the design problem of plate-fin heat exchanger, the geometric limitation of heat exchanger, such as shape and size, should be determined based on the actual engineering needs. Then the objective function is derived based on the design objective. The fluid data and constraints are brought into the operation, and the appropriate algorithm is selected for them. Finally, the optimal design data are obtained.

Bio-inspired optimization is based on a variety of physical and biology phenomena in the nature. Several typical examples include genetic algorithm (GA) based on evolutionary biology [46], particle swarm optimization (PSO) from simplified social models [25], simulated annealing algorithm from metallurgy for specific rate

✉ Zhile Yang
zl.yang@siat.ac.cn

1 Northeast Electric Power University, Jilin 132012, China

2 Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China

3 Intelligent Electrical Science and Technology Research Institute, Northeastern University, Shenyang 110819, China

4 State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources, North China Electric Power University, Beijing 102206, China

cooling operations [42], ant colony optimization (ACO) derived from the behavior of ants in food searching and paths discovering [13]. Moreover, a teaching learning-based optimization (TLBO) algorithm has also been proposed to simulate the teaching process of teachers and students [39].

In the heat exchanger optimal design optimization, many scholars have utilized bio-inspired optimization methods for solving the problems. The optimization of heat exchanger can be divided into two types: single objective and multi-objective. Featured optimization targets of optimal design include the amount of entropy produced, total heat exchange, maintenance costs and production costs. Port diameter, port horizontal distance, port vertical distance, plate thickness, fin length and number of plates are often used as geometric design variables [36]. Some recent proposed methods, such as Gaussian process regression and adaptive resampling, are adopted to minimize the cost [7]. Some scholars have combined traditional and novel methods, such as computational fluid dynamics (CFD) combination with GA, to optimize heat exchangers with Colburn factor $j$ and the friction factor $f$ as targets [2, 29]. Moreover, the optimal design problem is mostly solved by simple and improved optimization algorithm. Jonh and Krishnakumar [22] used GA to optimize the perforated plate matrix heat exchanger surfaces, where the optimal objectives were Colburn factor $j$ and the friction factor $f$. Rao and Saroj [38] used Jaya algorithm to optimize the shell and tube heat exchanger (STHE) economically, taking into account the consistency and maintainability caused by scaling. In the same year, Rao and Saroj [37] improved the Jaya algorithm and proposed the elitist-Jaya algorithm to optimize the setup and operation costs of the STHE. Dhavle et al. [12] used cohort intelligence (CI) to optimize STHE design, and the total cost of the heat exchanger was taken as the target, and the exterior scene of the tube and the spacing of baffles were taken as the design variables. Vasconcelos et al. [43] proposed a falcon optimization algorithm (FOA) based on the predation behavior of falcons. The total cost of STHE was used as the objective function. Iver et al. [21] combined GA and CI and proposed an adaptive range genetic algorithm (ARGA) to solve the costs of the economic optimization problems and extended them to solve other complicated core mechanical engineering application problems.

Sine cosine algorithm (SCA) is a new meta-heuristic algorithm first proposed by Mirjalili in 2016 [31]. This optimization method is based on sine and cosine functions in mathematics and has been immediately widely applied in various optimization problems. Ekiz et al. [14] expounded the feasibility of SCA in solving constraint optimization problems and concluded that the successful feature of SCA lies in the smooth balance of exploration

and development. Banerjee and Nabi [4] used SCA to optimize the shuttle's re-entry trajectory back to Earth to find the best trajectory during the return. Attia et al. [3] used the modified sine cosine algorithm (MSCA) to solve the problem of optimal power flow (OPF), which accelerated the speed of SCA search and avoided the local optimal selection of the original algorithm as much as possible. Hekimoglu [19] used SCA to solve the parameters of automatic voltage regulator and proportional integral derivative (PID). In the optimization problem of automatic voltage regulator system, SCA has strong robustness. Mahadad and Srairi [30] use SCA to optimize load margin stability to improve power system security. Wang et al. [44] employed the multi-objective sine cosine algorithm (MOSCA) combined with wavelet neutral network (WNN) to accurately predict the wind behavior of wind power plants. Das et al. used SCA to optimize the hydrothermal scheduling problem in the economics of power system to minimize the total cost of power generation and solve the short-term hydrothermal scheduling problem [11]. Chandrasekaran et al. [8] solved the partial shading detection (PSD) and MPPT (maximum power point tracking) problems using SCA.

Since the proposal of canonical SCA, many scholars have proposed various algorithm improvement schemes or combinations with other algorithms to solve the complex problems. Elaziz et al. [1] combined the opposition-based learning (OBL) idea with SCA to improve the search accuracy of SCA. Nenavath and Jatoth [32] combined SCA with differential evolution (DE) and demonstrated that the hybrid SCA-DE algorithm solves the problems effectively. Sindhu et al. [40] combined elite strategy with SCA to make simple improvements to the algorithm and improve the search efficiency of SCA. Bureerat and Pholdee [6] proposed adaptive sine cosine algorithm integrated with differential evolution (ASCA-DE) to solve the problem of structural damage detection. Gupta and Deep [17] proposed an improved SCA with crossover scheme and proved the validity using CEC2014 benchmark test. In addition, they also solved the multilevel threshold processing of image segmentation using the new method and obtained competitive result. Chegini et al. [9] combined the PSO and Levy flight methods and proposed the PSOSCALF with stronger jumping ability and higher searching ability. Gupta and Deep [16] added self-adaptive to SCA to enhance the exploitation ability. Pasandideh and Khalilpourazari [33] combined crow and SCA and proposed the sine cosine crow search algorithm to balance the exploration and exploitation. Li et al. [26] proposed the enhanced brain storm SCA (EBS-SCA) to solve the problem of premature convergence of SCA on complex optimization problem, which significantly enhanced the algorithm performance. Gupta and Deep [18] proposed

SCA with global optimization based on their previous studies, referred to simulated quenching algorithm search mechanism for improved algorithm, and used it to train multilayer perceptrons. Though a number of SCA variants have proposed, the optimization capacity of the algorithm is to be further improved, in particular toward the convergence speed aspect.

This paper tries to use dynamic opposite learning (DOL) to significantly enable the diversity of solutions in the SCA, where the elitist strategy is also used to accelerate the convergence. The DOL strategy was first proposed by Xu et al. [50], which significantly improved the comprehensive performance of TLBO . Combining DOL and elite policies with SCAs, a new EDOLSCA is proposed, taking the advantages of the adopted strategies to speed up SCA search and improve precision for specific problems. The ideas in this paper are evident in the CEC2014 benchmark test and PFHE design issues.

The major contribution of this article is summarized as follows:

Firstly, the objective function of PFHE optimization design is derived by using the method of number of transfer units (NTU), and two design objectives and fluid parameters are given.

Secondly, elite strategy and DOL are adopted to strengthen SCA, generating the EDOLSCA. The improvement makes the convergence speed of SCA greatly increase.

Finally, EDOLSCA was tested and compared using the CEC2014 benchmark and then adopted for solving an engineering example of PFHE optimal design.

The structure of this paper is as follows:

Section 2 demonstrates the thermodynamic formula of PFHE, establishes the optimization objective and evaluation function, as well as the constraint conditions. Sections 3 and 4 elaborate the preliminaries of the elite and DOL strategies and illustrate the principle and specific algorithm scheme of EDOLSCA. Section 5 provides a comprehensive test of EDOLSCA, including tests based on the CEC2014 benchmark and PFHE design, with comprehensive analysis of the results. Section 6 summarizes the whole paper and outlooks the future directions.

## 2 Previous work on plate-fin heat exchanger optimization

A standard two-flow straight finned plate-fin heat exchanger is shown in Fig.1. The plate-fin part of the plate-fin heat exchanger is shown in Fig.(a), in which the design variables mentioned below are marked. The fins are generally welded from thin aluminum plates. The picture shows a relatively simple form of fin. Figure(b) is the working principle diagram of the plate-fin heat exchanger. The two fluids flow alternately in different fins, exchanging heat with maximum efficiency. The optimization work in this paper is based on this featured heat exchanger. In this section, the formula is derived first; then, the optimization constraints and objectives are determined.

### 2.1 Formula derivation for plate-fin heat exchanger

Heat duty $Q$ is defined first, then $Q$ defines the overall energy exchange level of the heat exchanger, and the formula is given by Yousefi et al. [53]:

$$Q = \epsilon C_{min}(T_{Ain} - T_{Bin}) \tag{1}$$

$$C_{min} = m * C_p \tag{2}$$

where $C$ is the heat capacity rate, $T_{Ain}, T_{Bin}$ refer to fluid A (usually hot fluid) and fluid B (usually cold fluid), *in* refers to the fluid flowing into the heat exchanger, and *min* refers to the less numeric of the two fluids, *max* is the larger of the two fluids.

The effectiveness $\epsilon$ is given by Incropera et al. [20],

$$\epsilon = 1 - e^{\left(\left(\frac{1}{C_r}\right)NTU^{0.22}\left(e^{(-C_r NTU^{0.78})}-1\right)\right)} \tag{3}$$

where $e$ is the Eulers number,

$$C_r = C_{min}/C_{max} \tag{4}$$

Heat transfer coefficient method is a common method of thermodynamic solution, which is usually used to solve problems without knowing outlet temperature. The heat transfer coefficient is expressed as:

$$\frac{1}{NTU} =$$
$$C_{min}\left(\frac{A_{ffA}}{j_A Cp_A Pr_A^{-0.667} m_h A_A} + \frac{A_{ffB}}{j_B Cp_B Pr_B^{-0.667} m_B A_B}\right) \tag{5}$$

where $j$ is Colburn factor, $A, A_{HT}$ is heat transfer area (m$^2$), $Cp$ is specific heat of fluid (W· kg$^{-1}$ ·K$^{-1}$ ), $Pr$ is Prandtl number, and $N_s$ is number of entropy generation units.

The free flow area of two fluids is defined as:

$$Aff_A = (H_t - t_A)(1 - n_A t_A)L_B N_A \tag{6}$$

$$Aff_B = (H_B - t_B)(1 - n_B t_B)L_A N_B \tag{7}$$

In the formula, $H$ is height of the fin, $H$ is height of the fin, $t$ is fin thickness, $n$ is fin frequency, $L$ denotes heat exchanger length, $N$ is number of fin layers, $\mu$ is dynamic viscosity (N · s $^{-1}$ · m $^{-2}$), and $\rho$ is density (kg · m $^{-3}$).
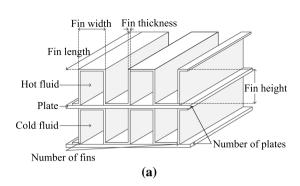
Fluid B has one more layer than fluid B:

**Fig. 1** Plate-fin heat exchanger and rectangular offset strip fin

$$N_B = N_A + 1 \tag{8}$$

The heat transfer area of the fluid comes from Yousefi [52]:

$$A_A = L_A L_B N_A (1 + (2n_A(H_A - t_A))) \tag{9}$$

$$A_B = L_A L_B N_A (1 + (2n_B(H_B - t_B))) \tag{10}$$

$j$ is the Colburn factor and $f$ is fanning friction factor [23].
For laminar flow $(Re \leq 1500)$,

$$j = 0.53(Re)^{-0.5} \left(\frac{l}{D_h}\right)^{-0.15} \left(\frac{s}{H} - t\right)^{-0.14} \tag{11}$$

$$f = 8.12(Re)^{-0.74} \left(\frac{l}{D_h}\right)^{-0.41} \left(\frac{s}{H} - t\right)^{-0.02} \tag{12}$$

For turbulent flow $(Re > 1500)$,

$$j = 0.21(Re)^{-0.4} \left(\frac{l}{D_h}\right)^{-0.24} \left(\frac{s}{H} - t\right)^{0.02} \tag{13}$$

$$f = 1.12(Re)^{-0.36} \left(\frac{l}{D_h}\right)^{-0.65} \left(\frac{s}{H} - t\right)^{0.17} \tag{14}$$

where $Re$ is Reynolds number, $s$ is fin spacing.
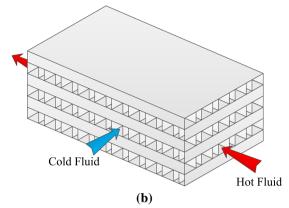The fin spacing of the PFHE:

$$s = (1/n - t) \tag{15}$$

where

$$Re = \frac{m \cdot D_h}{A_{ff}\mu} \tag{16}$$

For the fin shape in this paper, the hydraulic diameter $D_h$ of the heat exchanger can be expressed as:

$$D_h = \frac{4s(H - t)l}{2(sl + (H - t)l + t(H - t)) + ts} \tag{17}$$

where $D_h$ is hydraulic diameter and $l$ is lance length of the fin.

The pressure drop due to friction between the fluid in the channel and the fin can be calculated as:

$$\Delta P_A = \frac{2f_A L_A \left(\frac{m_A}{A_{ff_A}}\right)^2}{\rho_A D_{h,A}} \tag{18}$$

$$\Delta P_B = \frac{2f_B L_B \left(\frac{m_B}{A_{ff_B}}\right)^2}{\rho_B D_{h,B}} \tag{19}$$

where $\Delta P$ is pressure drop (N $\cdot$m$^{-2}$)

## 2.2 Objective function for plate-fin heat exchanger

For most heat exchangers, the flow of fluid in the channel is accompanied by friction, which leads to the process of entropy increase. The design of the heat exchanger determines the amount of entropy increase. Therefore, entropy increase is considered as an effective evaluation index in the design of heat exchanger. Under the same fluid condition, the smaller the number of entropy increases, the better the performance of the heat exchanger.

In this article, one of the target functions is defined as entropy generation units [5]:

$$Ns = (1 - \epsilon) \left[\frac{(T_{Bin} - T_{Ain})^2}{T_{Bin}T_{Ain}}\right] + \left(\frac{Rcte_A}{Cp_A}\right)\left(\frac{\Delta P_A}{P_{Ain}}\right) + \left(\frac{Rcte_B}{Cp_B}\right)\left(\frac{\Delta P_B}{P_{Bin}}\right) \tag{20}$$

where $Rcte$ is specific gas constant (J$\cdot$ kg$^{-1}$ $\cdot$K$^{-1}$)

In addition to entropy generation, the effective heat transfer area of heat exchanger is also one of the important indexes to evaluate heat exchanger. The heat transfer area is also closely related to the Reynolds number and efficiency of the heat exchanger. Therefore, the second objective function in this paper is the heat transfer area of the heat exchanger

The total heat transfer area is:

$$A_{HT} = A_A + A_B \tag{21}$$

## 2.3 Application example of plate-fin heat exchanger

For the two PFHE design examples prepared in this paper, the difference lies in the difference of fluid parameters and size constraints. The specific constraints and fluid data are shown in Table 1.

In order to meet the heat duty requirement of the heat exchanger, effective constraints are carried out in the calculation process of the algorithm, and the penalty function is used to achieve this requirement. The formula is as follows:

$$g(X) \Rightarrow \xi(X) - Q = 0 \tag{22}$$

In the design case, the overall size of the heat exchanger, the density of fins, the length, width and height of fins, the number of fluid layers of the two fluids and the thickness of fins are all the constraints of the design. The volume of the heat exchanger is limited to 1m*1m, and heat duty is 160kW of Case 1 and 1050kW of Case 2, respectively. In the calculations, fluids are considered ideal gases.

## 3 Algorithm preliminaries

Heuristic optimization algorithms are inspired by biology as well as natural sciences, such as physics and mathematics. Different ideas of particle updating determine the different characteristics of each algorithm. For different algorithm ideas, the parameters that need to be adjusted are also different. Some algorithms do not need to adjust parameters, while others may need to adjust 4–6 parameters to ensure the best results.

### 3.1 Sine cosine algorithm

For SCA, the main idea of the algorithm is the sine and cosine functions in mathematics. Four random numbers determine the motion direction and motion distance of particles, and there are no parameters to be adjusted in the algorithm. Therefore, SCA has relatively strong adaptability, which provides the possibility for wide application.

The core of the algorithm is as follows:

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \times sin(r_2) \times \left| r_3 P_t^i - X_i^t \right|, & r_4 < 0.5 \\ X_i^t + r_1 \times cos(r_2) \times \left| r_3 P_t^i - X_{i_i}^t \right|, & r_4 \geq 0.5 \end{cases} \tag{23}$$

$$r_1 = a - t\frac{a}{T} \tag{24}$$

**Table 1** Design constraints and fluid data

| Parameters | Project 1 | | Project 2 | |
|---|---|---|---|---|
| | Lower bound | Upper bound | Lower bound | Upper bound |
| Heat exchanger length , m, FluidA | 0.1 | 1 | 0.1 | 1 |
| Heat exchanger length , m, FluidB | 0.1 | 1 | 0.1 | 1 |
| Height of the fin, m | 0.002 | 0.01 | 0.002 | 0.01 |
| Fin frequency | 100 | 1000 | 100 | 1000 |
| Fin thickness, m | 0.0001 | 0.0002 | 0.0001 | 0.0002 |
| Lance length of the fin, m | 0.001 | 0.01 | 0.001 | 0.01 |
| Number of fin layers | 1 | 10 | 1 | 200 |

| Parameters | Project 1 | | Project 2 | |
|---|---|---|---|---|
| | Fluid A | Fluid B | Fluid A | Fluid B |
| Mass flow rate of fluid, $kg \cdot s^{-1}$ | 0.8962 | 0.8296 | 1.66 | 2 |
| Temperature, K | 513 | 277 | 1173.15 | 473.15 |
| Pressure, Pa | 100000 | 100000 | 160000 | 200000 |
| Specific heat of fluid, $J \cdot kg^{-1} \cdot K^{-1}$ | 1017.7 | 1011.8 | 1122 | 1073 |
| Density, $kg \cdot m^{-3}$ | 0.8196 | 0.9385 | 0.6296 | 0.9638 |
| Dynamic viscosity ,$N \cdot s \cdot m^{-2}$ | 2.410E-05 | 2.182E-05 | 4.010E-05 | 3.360E-05 |
| Prandtl number | 0.6878 | 0.6954 | 0.731 | 0.694 |
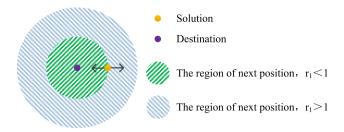
Fig. 2 Rules for the direction of particle motion in SCA

where $X$ is location of position, $t$ is the $t - th$ iteration , $P$ is final solution, and $T$ is maximum number of iterations. Since the main updating method in the formula involves sine and cosine function, it is named sine and cosine algorithm. The rules of the algorithm are shown more intuitively in Figs.2, 3. Figure 2 shows how the parameters in the equation affect the search space for the solution, which is a two-dimensional representation. Figures 2 and 3 together show the influence of sines and cosines on the next position in the calculation. The search space can be achieved by changing the range of values of sines and cosines. And the random position in space or out of space is also realized by equation parameters.In SCA, $r_1$ determines the direction of movement, $r_2$ determines the distance of movement, ranging in $[0,2\pi]$, $r_3$ determines the direction whether to move inward or outward, bringing randomness to the position, ranging in [-2,2], $r_4$ determines whether to move by sine or cosine manner, ranging in [0,1].

## 3.2 Dynamic-opposite learning

In some featured cases, optimization algorithm can be improved effectively by modifying certain steps, such as convergence speed and search accuracy. Oppositional-based learning (OBL) [35] is a popular modification method in recent years, from which quasi-opposite-based

learning (QOBL) [34] and quasi-reflection-based learning (QRBL) [15] are evolved.

OBL was first proposed to accelerate the convergence of DE. A strategy for finding the opposite number is proposed:

$$X^O = a + b - X \tag{25}$$

where $X^O$ is the opposite $X$, $X \in [a, b]$.

In practical application, if $D$ becomes the dimension of the problem, Eq.25 can be rewritten as:

$$X_j^O = a_j + b_j - X_j, j = 1 : D \tag{26}$$

In practical application, OBL, QOBL and QRBL are all likely to fall into the local optimal position in the search space. Xu et al. [50] proposed the dynamic opposite learning (DOL) strategy to solve this problem:

$$X^{DO} = X + w * rand * (rand * X^O - X) \tag{27}$$

where 'rand' is a random number in [0, 1] , $w$ is the weight that determines the combination of DOL and the original algorithm, that is, the weight of balance, and the 'rand * $X^O$' makes the search space become asymmetric, so as to solve the problem of falling into local optimal, and make the search become dynamic and constantly changing.

In the same way, DOL is applied to the real world and dimension $D$ is added, which can be rewritten as:

$$X_j^{DO} = X_j + w * rand * (rand * X_j^O - X_j), j = 1 : D \tag{28}$$

Applying the above DOL strategy to the initial population and iteration makes the algorithm search better and faster.

## 4 Elite and DOL-based SCA

In this section, we describe in details how EDOLSCA works. EDOLSCA combines elite and DOL policies to bring greater speed and precision to SCA.
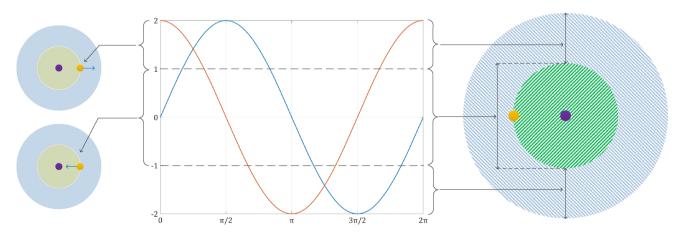


Fig. 3 Relationship between particle motion and sine and cosine functions in SCA

## 4.1 DOL population initialization

After the population is initialized, the first step will be calculated:

$$oP_{ij}^{DO} = oP_{ij} + r1_i * (r2_i * (a_j + b_j - oP_{ij}) - oP_{ij}) \quad (29)$$

where $oP$ is a random population generated within the upper and lower bound by using random numbers, and $i$ and $j$ are the $j^{th}$ variables of the $i^{th}$ individual. $a$ and $b$ are the boundaries of the variables, and $r_1$ and $r_2$ are random numbers. This step can improve the adaptability of the improved initial population through initial processing of the population. After modifying the population, the particle should be checked to see whether it exceeds the set boundary.

## 4.2 DOL generation jumping

After the DOL processes the initial population, the algorithm enters the SCA processing phase and is updated according to the SCA rules. In order to cooperate with different algorithms, jumping rate (Jr) is introduced into DOL as the basis for whether the algorithm enters into DOL step. Jr is a number in the range of [0, 1], and the specific value should be determined by experiments. In the algorithm step, the program randomly generates a random number in the range of [0, 1]. If the random number is less than the set Jr, then the algorithm will enter the DOL step:

$$P_{ij}^{DO} = P_{ij} + w * r3_i * (r4_i * (a_j + b_j - P_{ij}) - P_{ij}) \quad (30)$$

The boundary between A and B will be dynamically updated:

$$\begin{aligned} a_j &= min(P_{ij}) \\ b_j &= max(P_{ij}) \end{aligned} \quad (31)$$

As with the other steps, the updated population needs to check the boundaries.

## 4.3 EDOLSCA steps

In addition to the DOL strategy, the introduction of elites is intended to address the exploration speed. The concept of elite is to select a part of the best individuals before each iteration, skipping the intermediate steps and keeping until the end of each iteration to form a new population, so as to ensure the overall superiority of the population.

At this point, the specific process of EDOLSCA has been introduced, and the flowchart of the algorithm is shown in Fig. 4, and steps of EDOLSCA are shown in Algorithm.1.

---

**Algorithm 1** The EDOLSCA Algorithm

1: Generate initial population P;
2: **for** $i = 1; i \leq N; i++$ **do**
3:    $r1_i = rand(0,1), r2_i = rand(0,1)$;
4:    **for** $j = 1; j \leq D; j++$ **do**
5:      $P_{ij}^{DO} = P_{ij} + r1_i * (r2_i * (a_j + b_j - P_{ij}) - P_{ij})$ ;
6:      Check bound;
7:    **end for**
8: **end for**
9: Select $N$ number of the fittest individuals from $P \cup P^{DO}$;
10: Set G=1;
11: **while** $G <$ Maximum number of iterations **do**
12:    Evaluate all learners by the fitness function;
13:    **if** $G = 1$ **then**
14:      The best 0.05 of the initial population is selected as the elite;
15:    **else**
16:      Select 0.05 elites from the updated population;
17:    **end if**
18:    The selected elites will not participate in SCA and DOL calculations until the end of this iteration;
19:    **for** $i = 1; i \leq N; i++$ **do**
20:      $r_1 = a - t\frac{a}{T}$;
21:      $r_4 = rand[0,1]$;
22:      **if** $r_4 < 0.5$ **then**
23:        $X_i^{t+1} = X_i^t + r_1 \times sin(r_2) \times |r_3 P_t^i - X_i^t|$
24:      **else**
25:        $X_i^{t+1} = X_i^t + r_1 \times cos(r_2) \times |r_3 P_t^i - X_{i_i}^t|$
26:      **end if**
27:      Check bound;
28:      Assess the fitness of the new population $p'$ ;
29:      **if** $f(p)<f(p')$ **then**
30:        Replace $p$ with $p'$;
31:      **end if**
32:    **end for**
33:    $G++$;
34:    **if** $rand \leq Jr$ **then**
35:      **for** $i = 1; i \leq N; i++$ **do**
36:        $r3_i = rand(0,1), r4_i = rand(0,1)$;
37:        **for** $j = 1; j \leq D; j++$ **do**
38:          $a_j = min(P_{ij}), b_j = max(P_{ij})$;
39:          $P_{ij}^{DO} = P_{ij} + w*r3_i*(r4_i*(a_j+b_j-P_{ij})-P_{ij})$ ;
40:        Check the boundaries;
41:        **end for**
42:      **end for**
43:      Select $N$ fittest individuals from $P \cup P^{DO}$;
44:      $G++$;
45:    **end if**
46:    Remove the worst 0.05 of the population and add the elite;
47: **end while**

---

# 5 Experimental results and discussion

## 5.1 Benchmark test

In this test, the benchmark proposed by CEC2014 [27] was selected as the test function. The 14 alternative benchmarks are shown in Table 2, the first four benchmarks are unimodal functions, and the remaining twelve are multi-modal functions. These benchmarks are effective to test the exploration and development capabilities of algorithms. In
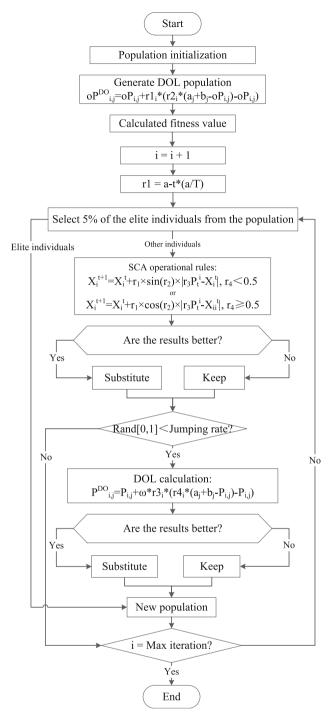
**Fig. 4** The flowchart of EDOLSCA

order to make the comparison of algorithms more intuitive, this paper selects four standard and classical algorithms with improvements, namely, PSO [25], cfPSO [10], cfwPSO and ETLBO [24]. Algorithm parameters are as follows: for PSO, $C_1 = C_2 = 2$; for cfPSO and cfwPSO,

$C_1 = C_2 = 2.05$, $K = 0.729$. For the parameters of EDOLSCA, w range is 1–15; Jr range is 0.1–1. The algorithms run 10 times for independent testing, and the results are shown in Tables 3, 4. The average with the smallest index is chosen, that is $w = 10$ and $Jr = 1$. The population size in the test was set to 50, and the number of iterations was 20,000. Results of the mean and standard deviation of each algorithm at each benchmark are recorded to compare the algorithms. Average index is considered as one of the criteria of comprehensive evaluation algorithm, which is calculated from the ranking of each algorithm. The calculated results are shown in Table 5, and the convergence diagram is shown in Fig.5.

First of all, it can be seen from the calculation results in the table that, for F1-F4, three of the four baselines EDOLSCA reach the optimal value, and in F5-F14, most EDOLSCA also get the minimum value of all algorithms. 'Ave index' shows that EDOLSCA has the best comprehensive result among all comparison algorithms, and only the improved SCA in one benchmark performs less well than the original algorithm. It is observed in the figure that EDOLSCA converges much faster than SCA in most benchmarks, which proves that the improvement is successful. In addition to using the average index method to detect EDOLSCA, the Wilcoxon rank-sum test [47] method is also introduced to analyze the results of the algorithm. The significance is set to 0.05, two-tailed. The test results are shown in Table 6.In the result table, '↑' denotes the EDOLSCA is better than other counterpart, '←' means other algorithms (the algorithm pointed by the arrow) is better, and '=' means there is no significant difference between the two algorithms after the test. The results show that the convergence accuracy has a small improvement compared with SCA, but the overall improvement ratio is not high. It shows that EDOLSCA has an absolute advantage in convergence speed, but it can still be improved in terms of accuracy.

## 5.2 Application to PFHE design problems

In this section, the optimization object is extended to the design problem of PFHE. In fact, the design optimization problem of PFHE is different from the repeatability optimization. The design optimization problem only needs an optimal design, and the stability of the repeatability optimization is more important [54]. Therefore, the stability of the algorithm is not considered in the PFHE problem, where only the accuracy of the algorithm is considered. For the heat transfer area objective function, both cases 1 and 2 are applicable. In order to optimize the environment

**Table 2** Unimodal and multimodal benchmark

| Function number | Function name |
|---|---|
| F1 | Rotated High Conditioned Elliptic Function |
| F2 | Rotated Bent Cigar Function |
| F3 | Rotated Discus Function |
| F4 | Shifted and Rotated Rosenbrock's Function |
| F5 | Shifted and Rotated Weierstrass Function |
| F6 | Shifted and Rotated Griewank's Function |
| F7 | Shifted Rastrigin's Function |
| F8 | Shifted and Rotated Rastrigin's Function |
| F9 | Shifted Schwefel's Function |
| F10 | Shifted and Rotated Schwefel's Function |
| F11 | Shifted and Rotated Katsuura Function |
| F12 | Shifted and Rotated HappyCat Function |
| F13 | Shifted and Rotated HGBat Function |
| F14 | Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function |

**Table 3** The sensitivity analysis of $w$

| Weight | Mean | | | | Ave index |
|---|---|---|---|---|---|
| | F1 | F3 | F4 | F7 | |
| $w = 1$ | 5.27E + 07 | 2.03E + 04 | 1.66E + 02 | 5.07E + 00 | 13.75 |
| $w = 2$ | 3.36E + 07 | 1.76E + 04 | 1.41E + 02 | 2.75E + 00 | 11 |
| $w = 3$ | 1.69E + 07 | 1.79E + 04 | 1.42E + 02 | 2.24E + 00 | 8.75 |
| $w = 4$ | 2.02E + 07 | 1.96E + 04 | 1.17E + 02 | 2.13E + 00 | 10 |
| $w = 5$ | 1.94E + 07 | 1.93E + 04 | 8.24E + 01 | 1.33E + 00 | 8.25 |
| $w = 6$ | 1.62E + 07 | 1.69E + 04 | 9.63E + 01 | 1.54E + 00 | 6.75 |
| $w = 7$ | 2.52E + 07 | 2.02E + 04 | 4.62E + 01 | 1.43E + 00 | 8.5 |
| $w = 8$ | 2.17E + 07 | **1.55E + 04** | 5.37E + 01 | 1.08E + 00 | 6.5 |
| $w = 9$ | 1.88E + 07 | 1.80E + 04 | 5.08E + 01 | 1.01E + 00 | 5.75 |
| $w = 10$ | 1.46E + 07 | 1.88E + 04 | 4.64E + 01 | 9.22E-01 | **4** |
| $w = 11$ | 1.72E + 07 | 2.04E + 04 | 5.59E + 01 | 9.09E-01 | 6.75 |
| $w = 12$ | 2.74E + 07 | 2.36E + 04 | 5.45E + 01 | 9.48E-01 | 9.5 |
| $w = 13$ | 2.10E + 07 | 2.48E + 04 | 4.41E + 01 | **8.97E-01** | 6.5 |
| $w = 14$ | 2.70E + 07 | 2.05E + 04 | **3.49E + 01** | 9.71E-01 | 7.5 |
| $w = 15$ | **1.44E + 07** | 2.73E + 04 | 4.61E + 01 | 1.05E + 00 | 6.5 |

Bold values indicate the best results compared to the other counterparts

**Table 4** The sensitivity analysis of $Jr$

| Jumping rate | Mean | | | | Ave index |
|---|---|---|---|---|---|
| | F1 | F3 | F4 | F7 | |
| $Jr = 0.1$ | 2.39E + 07 | 2.66E + 04 | 4.96E + 01 | 1.88E + 00 | 9.25 |
| $Jr = 0.2$ | 3.07E + 07 | 2.40E + 04 | 4.58E + 01 | 1.11E + 00 | 8 |
| $Jr = 0.3$ | 1.66E + 07 | 2.27E + 04 | 4.60E + 01 | 9.15E-01 | 7 |
| $Jr = 0.4$ | 1.10E + 07 | 1.90E + 04 | 5.05E + 01 | 6.57E-01 | 6.5 |
| $Jr = 0.5$ | 1.68E + 07 | 1.43E + 04 | 4.79E + 01 | 5.26E-01 | 6.25 |
| $Jr = 0.6$ | 1.35E + 07 | 1.28E + 04 | **3.41E + 01** | 3.43E-01 | 2.75 |
| $Jr = 0.7$ | 1.58E + 07 | 1.41E + 04 | 5.62E + 01 | 5.94E-01 | 6.5 |
| $Jr = 0.8$ | 9.57E + 06 | 1.30E + 04 | 4.42E + 01 | 6.08E-01 | 3.75 |
| $Jr = 0.9$ | 1.27E + 07 | **9.60E + 03** | 4.61E + 01 | 4.47E-01 | 3.5 |
| $Jr = 1$ | **6.42E + 06** | 1.13E + 04 | 4.42E + 01 | **3.92E-01** | **1.5** |

Bold value indicates the best results compared to the other counterparts

realistically, the number of iterations is set to 500,000, and the population is still 50. For the objective function of entropy increase, only test case 1, the number of iterations is set to 4,000, and the population is 50. In order to compare the fairness of the study, we add two varieties of algorithms in this section: elite sine cosine algorithm (ESCA), dynamic opposite learning sine cosine algorithm (DOLSCA) and two common algorithms: differential evolution (DE) [41] and neighborhood field for cooperative optimization (NFO) [48]. The results are shown in Figs.6, 7.

It can be seen from the result image that SCA and EDOLSCA have the same advantage in the *AHT* objective function. The EDOLSCA continued to converge under a high number of iterations, and the improved algorithm has higher accuracy than the original algorithm. On the *Ns* objective function, EDOLSCA converges faster than SCA, and the accuracy difference is larger. At the same time, in

**Table 5** The mean and standard value of unimodal/multi-modal test functions

| Algorithms | F1 | | F2 | | F3 | | F4 | |
|---|---|---|---|---|---|---|---|---|
| | Mean | Std | Mean | Std | Mean | Std | Mean | Std |
| PSO | 6.32E + 07 | 3.26E + 07 | 8.52E + 07 | 6.17E + 07 | **1.17E + 03** | 7.06E + 02 | 4.09E + 02 | 8.59E + 01 |
| cfPSO | 2.74E + 08 | 9.05E + 07 | 1.19E + 10 | 5.61E + 09 | 2.31E + 04 | 6.30E + 03 | 1.48E + 03 | 6.14E + 02 |
| cfwPSO | 2.29E + 08 | 8.71E + 07 | 1.51E + 10 | 5.50E + 09 | 3.24E + 04 | 8.70E + 03 | 2.00E + 03 | 8.59E + 02 |
| ETLBO | 3.84E + 08 | 1.29E + 08 | 3.54E + 10 | 8.50E + 09 | 6.58E + 04 | 1.38E + 04 | 3.93E + 03 | 8.02E + 02 |
| SCA | 2.82E + 07 | 1.09E + 07 | 5.04E + 08 | 5.22E + 08 | 4.73E + 04 | 1.28E + 04 | 1.19E + 02 | 5.88E + 01 |
| EDOLSCA | **2.63E + 07** | **2.08E + 07** | **7.38E + 07** | **1.10E + 08** | 5.11E + 04 | **2.51E + 04** | **1.03E + 02** | **5.05E + 01** |

| Algorithms | F5 | | F6 | | F7 | | F8 | |
|---|---|---|---|---|---|---|---|---|
| PSO | 2.66E + 01 | 2.08E + 00 | 2.55E + 00 | 1.41E + 00 | 9.31E + 01 | 1.88E + 01 | 1.16E + 02 | 2.35E + 01 |
| cfPSO | 2.66E + 01 | 2.81E + 00 | 1.56E + 02 | 6.13E + 01 | 1.29E + 02 | 2.72E + 01 | 1.45E + 02 | 2.84E + 01 |
| cfwPSO | 2.92E + 01 | 2.30E + 00 | 1.67E + 02 | 5.62E + 01 | 1.38E + 02 | 2.64E + 01 | 1.55E + 02 | 3.18E + 01 |
| ETLBO | 3.41E + 01 | 2.43E + 00 | 3.62E + 02 | 1.04E + 02 | 2.57E + 02 | 1.73E + 01 | 2.62E + 02 | 3.26E + 01 |
| SCA | **2.12E + 01** | 2.77E + 00 | 6.53E + 00 | 9.34E + 00 | 1.02E + 02 | 1.90E + 01 | 1.17E + 02 | 1.41E + 01 |
| EDOLSCA | 3.27E + 01 | **5.83E + 00** | **1.58E + 00** | **1.86E + 00** | **5.57E + 01** | **1.43E + 01** | **7.80E + 01** | **1.49E + 01** |

| Algorithms | F9 | | F10 | | F11 | | F12 | |
|---|---|---|---|---|---|---|---|---|
| PSO | 2.09E + 03 | 5.04E + 02 | **3.68E + 03** | 6.70E + 02 | 5.09E-01 | 1.59E-01 | 5.83E-01 | 1.26E-01 |
| cfPSO | 3.04E + 03 | 6.28E + 02 | 4.04E + 03 | 6.52E + 02 | 5.10E-01 | 1.04E-01 | 3.32E + 00 | 4.02E-01 |
| cfwPSO | **2.01E + 03** | 5.01E + 02 | 4.82E + 03 | 1.19E + 03 | **1.04E + 00** | 5.68E-01 | 3.65E + 00 | 3.98E-01 |
| ETLBO | 6.51E + 03 | 4.25E + 02 | 6.96E + 03 | 6.15E + 02 | 3.08E + 00 | 5.26E-01 | 5.32E + 00 | 6.99E-01 |
| SCA | 3.33E + 03 | 5.65E + 02 | 4.19E + 03 | 2.93E + 02 | 1.41E + 00 | 2.04E-01 | 5.37E-01 | 8.89E-02 |
| EDOLSCA | 3.86E + 03 | **1.45E + 03** | 4.60E + 03 | **1.67E + 03** | 2.21E + 00 | **3.80E-01** | 3.15E-01 | **8.13E-02** |

| Algorithms | F13 | | F14 | | Best num | Ave index |
|---|---|---|---|---|---|---|
| PSO | 3.23E-01 | 1.90E-01 | 4.78E + 01 | 1.89E + 01 | 2 | 2.36 |
| cfPSO | 5.94E + 01 | 2.01E + 01 | 8.63E + 02 | 1.01E + 03 | 0 | 3.86 |
| cfwPSO | 6.82E + 01 | 1.23E + 01 | 4.84E + 02 | 6.49E + 02 | 2 | 4.07 |
| ETLBO | 1.30E + 02 | 2.77E + 01 | 4.22E + 04 | 3.32E + 04 | 0 | 5.86 |
| SCA | 1.12E + 00 | 2.75E + 00 | 2.65E + 01 | 8.50E + 00 | 1 | 2.64 |
| EDOLSCA | **2.74E-01** | **4.60E-02** | **1.85E + 01** | **1.04E + 01** | **9** | **2.21** |

Bold value indicates the best results compared to the other counterparts

(a) High Conditioned Elliptic Function

(b) Bent Cigar Function

(c) Discus Function

(d) Rosenbrock's Function

(e) Weierstrass Function

(f) Griewank's Function

(g) Rastrigin's Function

(h) Rotated Rastrigin's Function

(i) Schwefel's Function

(j) Rotated Schwefel's Function

(k) Katsuura Function

(l) HappyCat Function

(m) HGBat Function

(n) Expanded Griewank's plus Rosenbrock's Function
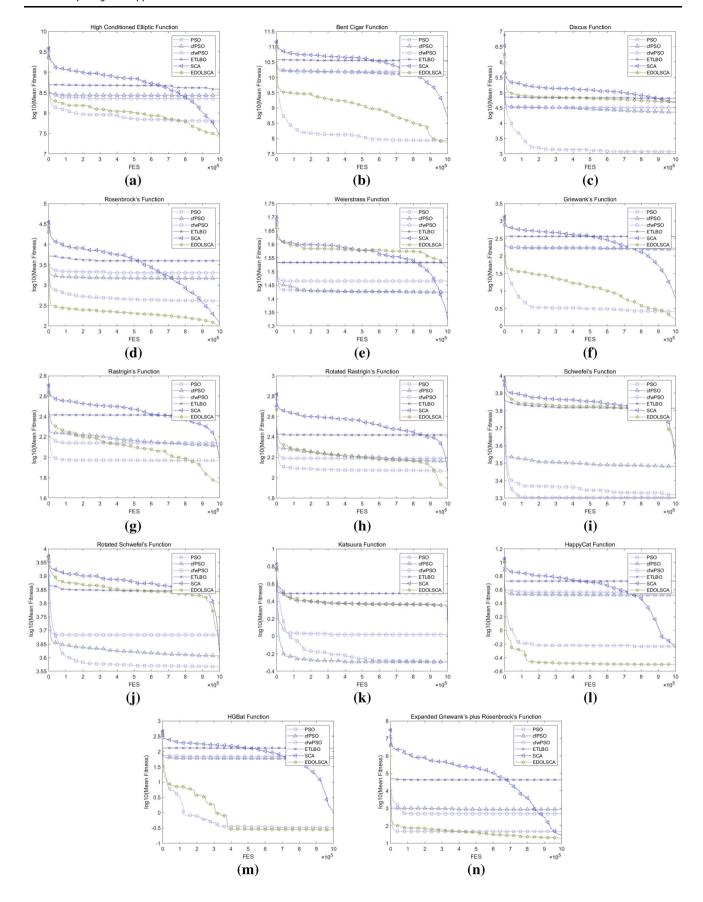
**Table 6** Wilcoxon rank-sum test on unimodal and multi-modal functions

| Algorithms | EDOLSCA | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | F1 | F2 | F3 | F4 | F5 | F6 | F7 | F8 |
| PSO | ↑ | = | ← | ↑ | ← | ↑ | ↑ | ↑ |
| cfPSO | ↑ | ↑ | = | ↑ | ← | ↑ | ↑ | ↑ |
| cfwPSO | ↑ | ↑ | = | ↑ | = | ↑ | ↑ | ↑ |
| ETLBO | ↑ | ↑ | = | ↑ | ↑ | ↑ | ↑ | ↑ |
| SCA | = | ↑ | = | = | ← | ↑ | ↑ | ↑ |

| Algorithms | EDOLSCA | | | | | | Same | Better |
|---|---|---|---|---|---|---|---|---|
| | F9 | F10 | F11 | F12 | F13 | F14 | | |
| PSO | ← | = | ← | ↑ | ↑ | ↑ | 2 | 8 |
| cfPSO | = | = | ← | ↑ | ↑ | ↑ | 3 | 9 |
| cfwPSO | ← | = | ← | ↑ | ↑ | ↑ | 3 | 9 |
| ETLBO | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | 1 | 13 |
| SCA | = | = | = | ↑ | = | = | 8 | 5 |

the three cases, the accuracy of EDOLSCA is the highest, which proves its potentials in engineering design optimization problems. The results of benchmark and engineering tests show that EDOLSCA has excellent dynamic search capabilities and has unique advantages in speed and accuracy, as well as demonstrating excellent comprehensive performance.
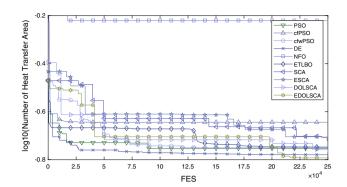


**Fig. 7** The solution result of the *Ns* objective, cases 1

# 6 Conclusion

This paper proposed a new EDOLSCA adopted elite and DOL strategies to strengthen the canonical SCA, which greatly improves the performance of the original algorithm in terms of convergence accuracy and convergence speed. The proposed method was compared with the state-of-the-art algorithms counterparts in the CEC2014 benchmark test proved to perform well. The algorithm is further adopted in solving the optimal design problem of PFHE, where the dynamic and random search space of the DOL strategy has achieved excellent results. The proposed algorithm again demonstrated superiority in solving the complex optimal design problem. Future work lies on the wide application of the proposed EDOLSCA in various industrial and science optimization scenarios. The well robustness of the algorithm is promising to provide a useful computational tool for engineering optimization.
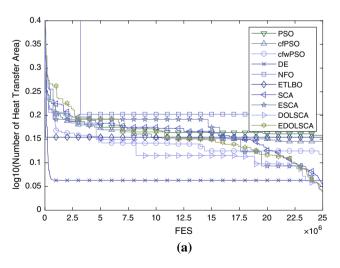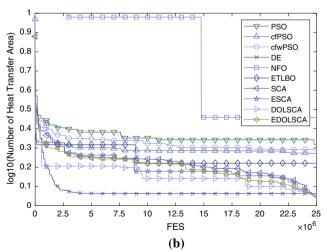


**Fig. 6** The solution result of the *AHT* objective, cases 1 and 2

## Declarations

## References

1. Abd Elaziz M, Oliva D, Xiong S (2017) An improved opposition-based sine cosine algorithm for global optimization. Exp Syst Appl 90:484–500
2. Abeykoon C (2020) Compact heat exchangers-design and optimization with cfd. Int J Heat Mass Trans 146:118766
3. Attia AF, El Sehiemy RA, Hasanien HM (2018) Optimal power flow solution in power systems using a novel sine-cosine algorithm. Int J Electric Power Energy Syst 99:331–343
4. Banerjee A, Nabi M (2017) Re-entry trajectory optimization for space shuttle using sine-cosine algorithm. In: 2017 8th international conference on recent advances in space technologies (RAST), pp. 73–77. IEEE
5. Bejan A (2013) Entropy generation minimization: the method of thermodynamic optimization of finite-size systems and finite-time processes. CRC Press
6. Bureerat S, Pholdee N (2017) Adaptive sine cosine algorithm integrated with differential evolution for structural damage detection. In: International Conference on Computational Science and Its Applications, pp. 71–86. Springer
7. Campet R, Roy PT, Cuenot B, Riber É, Jouhaud JC (2020) Design optimization of an heat exchanger using gaussian process. Int J Heat Mass Trans 150:119264
8. Chandrasekaran K, Sankar S, Banumalar K (2020) Partial shading detection for pv arrays in a maximum power tracking system using the sine-cosine algorithm. Energy Sustain Develop 55:105–121
9. Chegini SN, Bagheri A, Najafi F (2018) Psoscalf: A new hybrid pso based on sine cosine algorithm and levy flight for solving optimization problems. Appl Soft Comput 73:697–726
10. Clerc M, Kennedy J (2002) The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans Evol Comput 6(1):58–73
11. Das S, Bhattacharya A, Chakraborty AK (2018) Solution of short-term hydrothermal scheduling using sine cosine algorithm. Soft Comput 22(19):6409–6427
12. Dhavle SV, Kulkarni AJ, Shastri A, Kale IR (2018) Design and economic optimization of shell-and-tube heat exchanger using cohort intelligence algorithm. Neural Comput Appl 30(1):111–125
13. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. IEEE Comput Intell Mag 1(4):28–39
14. Ekiz S, Erdoğmuş P, Özgür B (2017) Solving constrained optimization problems with sine-cosine algorithm. Periodicals of Engineering and Natural Sciences 5(3)
15. Ergezer M, Simon D, Du D (2009) Oppositional biogeography-based optimization. In: 2009 IEEE international conference on systems, man and cybernetics, pp. 1009–1014. IEEE
16. Gupta S, Deep K (2019) A hybrid self-adaptive sine cosine algorithm with opposition based learning. Exp Syst Appl 119:210–230
17. Gupta S, Deep K (2019) Improved sine cosine algorithm with crossover scheme for global optimization. Knowledge-Based Syst 165:374–406
18. Gupta S, Deep K (2020) A novel hybrid sine cosine algorithm for global optimization and its application to train multilayer perceptrons. Appl Intell 50(4):993–1026
19. Hekimoğlu B (2019) Sine-cosine algorithm-based optimization for automatic voltage regulator system. Trans Ins Measure Control 41(6):1761–1771
20. Incropera FP, Lavine AS, Bergman TL, DeWitt DP (2007) Fundamentals of heat and mass transfer. Wiley
21. Iyer VH, Mahesh S, Malpani R, Sapre M, Kulkarni AJ (2019) Adaptive range genetic algorithm: a hybrid optimization approach and its application in the design and economic optimization of shell-and-tube heat exchanger. Eng Appl Artif Intell 85:444–461
22. John AK, Krishnakumar K (2017) Performing multiobjective optimization on perforated plate matrix heat exchanger surfaces using genetic algorithm. Int J Simul Multidiscip Design Optimiz 8:A3
23. Joshi HM, Webb RL (1987) Heat transfer and friction in the offset stripfin heat exchanger. Int J Heat Mass Trans 30(1):69–84
24. Kadambur R, Kotecha P (2015) Multi-level production planning in a petrochemical industry using elitist teaching-learning-based-optimization. Exp Syst Appl 42(1):628–641
25. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95-International Conference on Neural Networks, vol. 4, pp. 1942–1948. IEEE
26. Li C, Luo Z, Song Z, Yang F, Fan J, Liu PX (2019) An enhanced brain storm sine cosine algorithm for global optimization problems. IEEE Access 7:28211–28229
27. Liang JJ, Qu BY, Suganthan PN (2013) Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore **635**
28. Lincoln GA (1973) Energy conservation. Science 180(4082):155–162
29. Liu C, Bu W, Xu D (2017) Multi-objective shape optimization of a plate-fin heat exchanger using cfd and multi-objective genetic algorithm. Int J Heat Mass Trans 111:65–82
30. Mahdad B, Srairi K (2018) A new interactive sine cosine algorithm for loading margin stability improvement under contingency. Electric Eng 100(2):913–933
31. Mirjalili S (2016) Sca: a sine cosine algorithm for solving optimization problems. Knowledge-based Syst 96:120–133
32. Nenavath H, Jatoth RK (2018) Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking. Appl Soft Comput 62:1019–1043

33. Pasandideh SHR, Khalilpourazari S (2018) Sine cosine crow search algorithm: a powerful hybrid meta heuristic for global optimization. arXiv preprint arXiv:1801.08485

34. Rahnamayan S, Tizhoosh HR, Salama MM (2007) Quasi-oppositional differential evolution. In: 2007 IEEE congress on evolutionary computation, pp. 2229–2236. IEEE

35. Rahnamayan S, Tizhoosh HR, Salama MM (2008) Opposition-based differential evolution. IEEE Trans Evol Comput 12(1):64–79

36. Raja BD, Jhala R, Patel V (2018) Thermal-hydraulic optimization of plate heat exchanger: A multi-objective approach. International J Therm Sci 124:522–535

37. Rao RV, Saroj A (2017) Constrained economic optimization of shell-and-tube heat exchangers using elitist-jaya algorithm. Energy 128:785–800

38. Rao RV, Saroj A (2017) Economic optimization of shell-and-tube heat exchanger using jaya algorithm with maintenance consideration. Appl Therm Eng 116:473–487

39. Rao RV, Savsani VJ, Vakharia D (2012) Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems. Information sciences 183(1):1–15

40. Sindhu R, Ngadiran R, Yacob YM, Zahri NAH, Hariharan M (2017) Sine-cosine algorithm for feature selection with elitism strategy and new updating mechanism. Neural Comput Appl 28(10):2947–2958

41. Storn R, Price K (1997) Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optimiz 11(4):341–359

42. Van Laarhoven PJ, Aarts EH (1987) Simulated annealing. In: Simulated annealing: Theory Appl, pp. 7–15. Springer

43. de Vasconcelos Segundo EH, Mariani VC, dos Santos Coelho L (2019) Design of heat exchangers using falcon optimization algorithm. Appl Therm Eng 156:119–144

44. Wang J, Yang W, Du P, Niu T (2018) A novel hybrid forecasting system of wind speed based on a newly developed multi-objective sine cosine algorithm. Energy Convers Manag 163:134–150

45. Wang Z, Li Y (2016) Layer pattern thermal design and optimization for multistream plate-fin heat exchangers–a review. Renew Sustain Energy Rev 53:500–514

46. Whitley D (1994) A genetic algorithm tutorial. Statist Comput 4(2):65–85

47. Wilcoxon F(1992) Individual comparisons by ranking methods. In: Breakthroughs in statistics, pp. 196–202. Springer

48. Wu Z, Chow TW (2013) Neighborhood field for cooperative optimization. Soft Comput 17(5):819–834

49. Wu Z, Li Q, Wu W, Zhao M (2019) Crowdsourcing model for energy efficiency retrofit and mixed-integer equilibrium analysis. IEEE Trans Ind Inform 16(7):4512–4524

50. Xu Y, Yang Z, Li X, Kang H, Yang X (2020) Dynamic opposite learning enhanced teaching-learning-based optimization. Knowledge-Based Syst 188:104966

51. Yang Z, Mourshed M, Liu K, Xu X, Feng S (2020) A novel competitive swarm optimized rbf neural network model for short-term solar power generation forecasting. Neurocomputing 397:415–421

52. Yousefi M, Darus A, Mohammadi H (2012) An imperialist competitive algorithm for optimal design of plate-fin heat exchangers. Int J Heat Mass Transf 55(11–12):3178–3185

53. Yousefi M, Enayatifar R, Darus AN, Abdullah AH (2013) Optimization of plate-fin heat exchangers by an improved harmony search algorithm. Appl Therm Eng 50(1):877–885

54. Yuen SY, Lou Y, Zhang X (2019) Selecting evolutionary algorithms for black box design optimization problems. Soft Comput 23(15):6511–6531

55. Zhang L, Li Y, Zhang H, Xu X, Yang Z, Xu W (2021) A review of the potential of district heating system in northern china. Appl Therm Eng p. 116605