IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 6, 2020, accepted June 20, 2020, date of publication July 8, 2020, date of current version August 5, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3007602

An Improved Grasshopper Optimization Algorithm for Optimizing Hybrid Active Power Filters' Parameters

JUNRU HUANG, CHUNQUAN LI, (Member, IEEE), ZHILING CUI, LEYINGYUE ZHANG, AND WANXUAN DAI
School of Information Engineering, Nanchang University, Nanchang 330029, China

Corresponding author: Chunquan Li (lichunquan@ncu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61503177, Grant 81660299, and Grant 61863028, in part by the China Scholarship Council under the State Scholarship Fund under Grant CSC 201606825041, in part by the Science and Technology Department of Jiangxi Province of China under Grant 20161ACB21007, Grant 20171BBE50071, and Grant 20171BAB202033, and in part by the Education Department of Jiangxi province of China under Grant GJJ14228 and Grant GJJ150197.

ABSTRACT The selection of parameters for the hybrid active power filter (HAPF) is essential to harmonic compensation. To optimize HAPF parameters, this paper presents an improved grasshopper optimization algorithm (IGOA). In the IGOA, the whole population is divided into two sub-populations which focus on exploration and exploitation respectively. An improved social interaction mechanism is proposed to balance global and local searches. Furthermore, a learning strategy is introduced and an exemplar pool is built to replace the target in the original GOA, which can enhance the global search ability and escape local optima. The proposed IGOA is employed to optimize the parameters of two prevalent HAPF topologies for several cases. The experimental results show that the IGOA can get a promising performance compared with previous studies and other meta-heuristic algorithms.

INDEX TERMS Grasshopper optimization algorithm (GOA), hybrid active power filter (HAPF), harmonic pollution, computational intelligence.

I. INTRODUCTION

As the power electronic technology develops, many non-linear loads and rectifiers have been applied widely in the industry. These devices brought a negative impact on power quality. Among these, significant amounts of harmonics were introduced to the power grid, which causes severe degradation of power quality. Thus, research of harmonic elimination is of great significance.

To alleviate the effects of harmonic pollution, passive power filters (PPFs), active power filters (APFs), and hybrid active power filters (HAPFs) are widely used in the power grid. Basically, passive filter is a passive circuit network composed of inductors, capacitors, and resistors. The parameters of the element and the topology of the circuit affect the frequency characteristics of the network used to eliminate harmonics. A passive filter is an economical approach that easy to design and implement. However, only specific frequencies of harmonics can be filtered, which is lack of flexibility as the parameters of its component are relatively fixed. Active power filters (APF) usually detect harmonic currents and counteract them with a controlled current source. Unfortunately, active filters have not been widely applied due to its high economic cost. In recent years, hybrid active power filters (HAPF), which combine active filter and passive filter, have become an effective approach to suppress harmonic [1]–[3]. In HAPF, the passive filter acts dominantly in harmonic compensation whilst active filter improves the filtering characteristic of PPF and suppresses resonance. This structure can effectively decrease the capacitor and rating of the active filter, thereby solve the high- cost issue. However, the design of the HAPF is more complicated. For example, the harmonic current flows through passive filters then produced harmonic voltage, which affects the capacity of the active filter. Meanwhile, the harmonic filter design must achieve the best solution while satisfying many constraints, such as power factors and cost-effectiveness. Consequently, the parameter selection of passive filter (L and C) and active filter (gain value G) is of great significance.

Essentially, the design of HAPF and PPF is a complex non-linear programming problem that can be considered as an optimization issue. Various studies had been carried

The associate editor coordinating the review of this manuscript and approving it for publication was Sun-Yuan Hsieh.

out, and many single objective and multi-objective models had been established for designing power filters. Reference [4] constructed a multi-objective model for designing large scale PPF and implemented hybrid differential evolution (HDE) to design the parameters. Reference [5]–[7] utilized the genetic algorithm (GA) to optimize the parameters. However, GA contains selection, copy, crossover and mutation, and other processes, which make it complex to implement. Reference [8] presented Mixed Integer Distributed Ant Colony Optimization (MIDACO) for solving multi-objective single-tuned passive filter design problems. Reference [9] presented bacterial foraging optimization (BFO) algorithm and its adaptive version to optimize the planning of passive harmonic filters. Reference [10] presented the Harris hawks optimization (HHO) algorithm to obtain the optimal parameters of passive filters.

Most previous researches aimed at the optimal design of passive filters, while the application of meta-heuristic algorithms in hybrid active power filter (HAPF) remains scarce. Reference [11] constructed a multi-objective model for HAPF and applied PSO to search the optimal solution of the filters. Zobba [12] proposed a multi-objective model and utilized FORTRAN Feasible Sequential Quadratic Programming (FFSQP) to determine the optimal design of hybrid filters. Suganthan [13] formulated a single objective function and proposed L-SHADE to optimize the parameters of HAPF. Compared with reference [12], it significantly increased the chance to find the optimal solution and improved the performance of harmonic compensation. Despite the success of the proposed L-SHADE for obtaining the best parameters to minimize the harmonic pollution, it still suffers from being trapped into local optima and low average performance owing to the solution space with local optima. Therefore, we carry out research to improve the performance of parameter extraction based on the objective function proposed by Suganthan [13].

The Grasshopper optimization algorithm (GOA) is a novel meta-heuristic algorithm proposed by Mirjalili *et al.* [14]. It has been applied in various areas since its proposal for its fast convergence speed and highly accurate solutions. Wu [15] proposed adaptive GOA (AGOA) for trajectory optimization method for multiple Solar-powered unmanned aerial vehicle (SUAV) target tracking in urban environment. Luo *et al.* [16] proposed Improved GOA and applied it successfully to the financial stress prediction problem. Elmi *et al.* [17] presented a new algorithm for robot path planning in a static environment. El-Fergany [18] applied GOA to obtain optimum values of unknown seven parameters of proton exchange membrane fuel cells stack.

However, as a young population-based stochastic optimization algorithm, GOA has several disadvantages of easily felling into local optima and poor global search ability, which prevent its further application in more areas. Various studies had been carried out to further improve and explore its performance. Wu *et al.* [15] introduced some improvement measures such as natural selection strategy and the dynamic feedback mechanism to GOA and made an effective solution on the specific problem. Ewees [19] applied the oppositional-based learning strategy in GOA to obtain a more sufficient search space. Arora [20] employed chaotic maps to GOA to balance the exploration and exploitation in the optimization process. The result proved that it significantly boosts the performance of GOA. However, among these GOA variants, there is no study applied GOA for solving the HAPF model's parameter optimizations.

This paper proposed an improved GOA(IGOA) for optimizing the parameters of the hybrid active power filter (HAPF). First, inspired by HCLPSO [21], we divide the whole swarm population into two subpopulations that focus on exploitation and exploration, respectively. Second, we introduce the learning strategy in ITLBO [22] to expand particles' search space and escape local optima. We also build an exemplar pool to replace the gbest part in the original GOA. Furthermore, a non-linear reduced coefficient c was adopted to optimize the convergence speed in different iterative stages.

Moreover, we optimize the social interaction mechanism to balance the exploration and exploitation and reduce the computational complexity at the same time. To assess the performance of our improvement, we compared the proposed algorithm with L-SHADE and several other meta-heuristic algorithms on the parameter extraction of HAPF models. The experimental results demonstrate that IGOA can achieve highly competitive results and outperform GOA and L-SHADE, as well as other state-of-art algorithms.

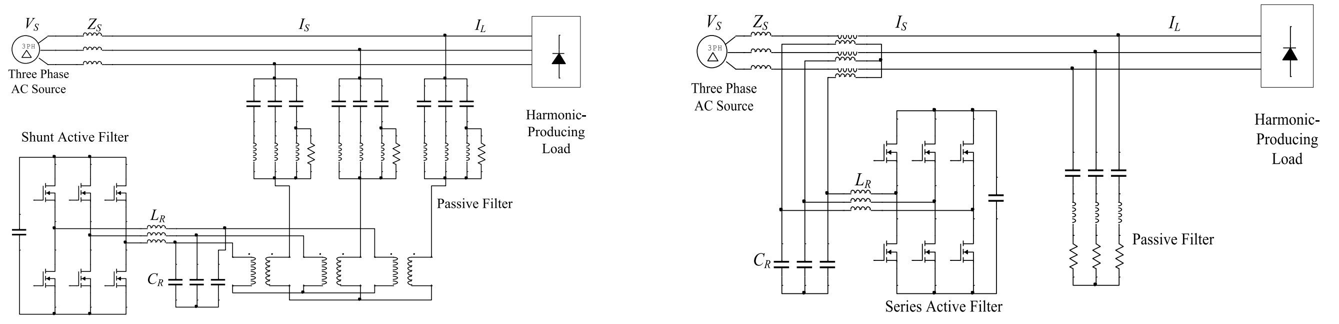
The main contributions of this paper are as follows:

(1) Different from other existing GOA variants, this paper optimized the social interaction mechanism, which fully utilizes the characteristic of s function to balance exploration and exploitation. Meanwhile, the computational complexity is significantly reduced.

(2) To the best of our knowledge, there is no GOA variant applied to optimize the parameter extraction of HAPF models. The existing L-SHADE algorithm has an unsatisfactory performance on minimizing harmonic pollutions for HAPF models. The proposed IGOA provides highly competitive results in parameter extraction based on the same objective function of the HAPF model in reference [13].

(3) An exemplar pool is built to replace the $\hat{T_d}$(target) in the original GOA, which can significantly boost the global search ability and avoid pre-mature. A multi-group search strategy was firstly applied in GOA, which enhanced the balance of exploration and exploitation for the particle swarm.

The remains of this paper are organized as follows: Section II illustrates the HAPF circuit and model formulation. Section III describes the original GOA algorithm. Section IV presents the proposed IGOA in detail. The experimental results and analysis of several cases are given in Section V.



(a) Config.1: APF in series with shunt passive filter

(b) Config.2: Combined series APF and shunt passive filter

FIGURE 1. Circuit configurations of HAPF.

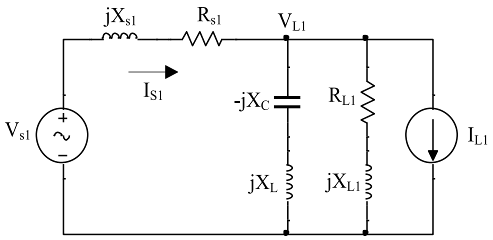


FIGURE 2. Single-phase equivalent circuit at fundamental frequency (n = 1).

II. HAPF: CIRCUIT ANALYSIS AND MODEL FORMULATION

In this paper, two frequently used topologies of HAPF for harmonic compensation are analyzed: APF in series with shunt passive filter and Combined series APF and shunt passive filter.

As Fig.1 illustrates, they are the series or parallel combination of three-phase three-wire active filter and passive filter. The difference between them lies in the location of the active filter and its function, which will be presented later.

The series APF acts as a controllable voltage source ($v_c = GI_{sn}$) that controlled by I_{sn} (source harmonic current), where G is the controllable gain of the filter. The series APF acts as a harmonic isolator that presents zero impedance to external circuit at the fundamental frequency and high resistance at harmonic frequency [23], [24]. Hence, the equivalent circuit for two topologies is identical at the fundamental frequency, as Fig. 2 indicates.

The Thevenin voltage source representing the utility supply voltage and the harmonic current source representing the non-linear load are [25]:

$$v_s(t) = \sum_n v_{Sn}(t) \tag{1}$$

$$i_L(t) = \sum_n i_{Ln}(t) \tag{2}$$

The *n-th* harmonic Thevenin source impedance is:

$$Z_{Sn} = R_{Sn} + jX_{Sn} \tag{3}$$

and the *n-th* harmonic load impedance is:

$$Z_{Ln} = R_{Ln} + jX_{Ln} \tag{4}$$

the *n-th* admittance

$$Y_{Ln} = G_{Ln} - jB_{Ln} \tag{5}$$

where G_{Ln}, B_{Ln} is the load conductance and susceptance in mho at harmonic number 'n';R_{Sn}, X_{Sn} are the transmission system resistance and reactance in ohms at harmonic number 'n'.

Analysis of the equivalent circuit is presented as follows:

A. APF IN SERIES WITH SHUNT PASSIVE FILTER

In this configuration, a small-rated series active filter is connected in series with a passive filter. The active filter injects all the harmonic currents into the passive filter to cancel the load harmonics and provides fundamental current so that no harmonic current flows in the source. The function of the active filter is to solve the problems inherent in using the passive filter alone [26]. The compensated utility supply current and load voltage are obtained via formula (6), (7) respectively.

$$I_{Sn} = \frac{A + jB}{C + jD} \tag{6}$$

$$V_{Ln} = \frac{E + jF}{C + jD} \tag{7}$$

where,

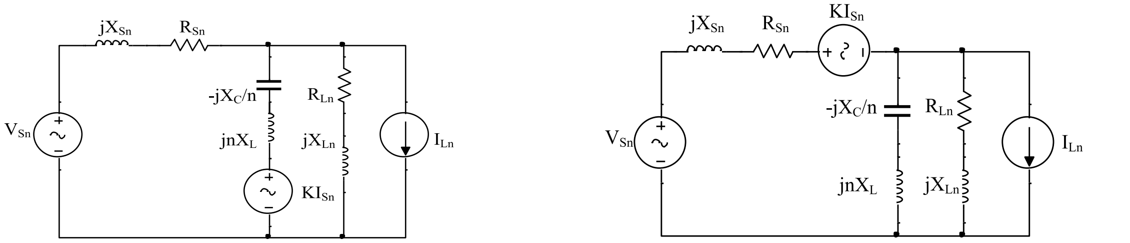
$$A = V_{Sn}R_{Ln} - I_{Ln}X_{Ln}X_{Fn} \tag{8}$$

$$B = V_{Sn}(X_{Ln} + X_{Fn}) + I_{Ln}R_{Ln}X_{Fn} \tag{9}$$

$$C = R_{TLn} + GR_{Ln} - (X_{Ln} + X_{Sn})X_{Fn} \tag{10}$$

$$D = X_{TLn} + GX_{Ln} + (R_{Ln} + R_{Sn})X_{Fn} \tag{11}$$

$$\begin{aligned} E = {} & V_{Sn}\left(GR_{Ln} - X_{Ln}\left(nX_L - \frac{X_c}{n}\right)\right) \\ & + I_{Ln}X_{TLn}\left(nX_L - \frac{X_c}{n}\right) \end{aligned} \tag{12}$$



(a) Config.1 'APF in series with shunt passive filter'

(b) Config.2 'Combined series APF and shunt passive filter'

FIGURE 3. **Single-phase equivalent circuit at harmonic frequencies (n ≥2).**

$$F = V_{Sn}\left(R_{Ln}\left(nX_L - \frac{X_c}{n}\right) + GX_{Ln}\right) - I_{Ln}R_{TLn}\left(nX_L - \frac{X_c}{n}\right) \quad (13)$$

so that:

$$R_{TLn} = R_{Sn}R_{Ln} - X_{Ln}X_{Sn} \quad (14)$$

$$X_{TLn} = R_{Ln}X_{Sn} + R_{Sn}X_{Ln} \quad (15)$$

B. COMBINED SERIES APF AND SHUNT PASSIVE FILTER

This topology was first proposed by Peng, F.Z. et.al [27]. As Fig. 3(b) indicates, the active filter is connected in series with the power line and the passive filter is connected in parallel to the power line. The high resistance at the harmonic frequency in the power line forces the harmonic current to flow through the passive filter. The compensated utility supply current and load voltage are calculated by

$$I_{Sn} = \frac{A + jB}{C + jD'} \quad (16)$$

$$V_{Ln} = \frac{E + jF'}{C + jD'} \quad (17)$$

where,

$$D' = X_{TLn} + GX_{Ln} + (R_{Ln} + R_{Sn} + G)\left(nX_L - \frac{X_c}{n}\right) \quad (18)$$

$$F' = V_{Sn}\left((R_{Ln} + G)\left(nX_L - \frac{X_c}{n}\right) + GX_{Ln}\right) - I_{Ln}R_{TLn}\left(nX_L - \frac{X_c}{n}\right) \quad (19)$$

the other system performance is formulated as follows:

The compensated load power factor (PF):

$$PF = \frac{P_L}{V_L I_S} = \frac{\sum_n G_{Ln}V_{Ln}^2}{\sqrt{\sum_n I_{Sn}^2 \sum_n V_{Ln}^2}} \quad (20)$$

Also, the compensated load displacement power factor (Dpf):

$$Dpf = \frac{P_L}{V_{L1}I_{S1}} \quad (21)$$

The transmission losses (P_{Loss}):

$$P_{Loss} = \sum_n I_{Sn}^2 R_{Sn} \quad (22)$$

The transmission efficiency (η):

$$\eta = \frac{P_L}{P_{Loss} + P_L} = \frac{\sum_n G_{Ln}V_{Ln}^2}{\sum_n I_{Sn}^2 R_{Sn} + \sum_n G_{Ln}V_{Ln}^2} \quad (23)$$

Compensated VTHD at the load terminal

$$VTHD = \frac{\sqrt{\sum_{n\geq 2} v_{Ln}^2}}{v_{L1}} \quad (24)$$

Compensated ITHD for the utility supply current,

$$ITHD = \frac{\sqrt{\sum_{n\geq 2} I_{Ln}^2}}{I_{L1}} \quad (25)$$

According to [28], the formula of harmonic pollution is:

$$HP = \sqrt{VTHD^2 + ITHD^2} \quad (26)$$

C. OBJECTIVE FUNCTION

The major goal for HAPF model's parameter estimation is to discover a set of the optimal value of filters to reduce the harmonic pollution. In this article, the HAPF parameters to be optimized are: X_c(capacitive reactance), X_L (inductive reactance) and G (controllable gain of the active filter), which are complying with the following constraints:

$$0 \leq X_c \leq 10$$

$$0 \leq X_L \leq 1$$

$$0 \leq G \leq 20$$

The author in reference [13] proposed the objective function that formulated as follows:

$$\text{Maximize } 'HP'_{APP} \quad \text{subject to } PF = PF_{goal} \pm \varepsilon$$

where,

$$HP_{APP} = abs\,(VTHD_{lim} - VTHD) + abs\,(ITHD_{lim} - ITHD) \quad (27)$$

$VTHD_{lim}$ = limitation on $VTHD$ prescribed by IEEE 519-2014 [28] based on system voltage level.

$ITHD_{lim}$ = limitation on $ITHD$ prescribed by IEEE 519-2014 [28] based on system short circuit ratio.

PF_{goal} is the desired power factor, which is set to 95% in this work. ε is a very small error value set to 1% to facilitate the iteration process.

IGOA performs minimization of an objective function while the objective function of this paper (Eq. 27) is to maximize 'HP_{APP}'. Therefore, the '$-HP_{APP}$' (negative) is input as the objective function that be minimized by selecting three parameters: X_L, X_c and G. As the HP_{APP} is maximized, the *VTHD* and *ITHD* move towards 0, thereby reduce the HP value. Note that the HP value is calculated by Eq. 26 and directly linked to the performance of the harmonic filters. The less HP value is, the better the compensating effect of the filters are. After the optimization, the optimal set of parameters will lead to the minimization of the HP value. Therefore, the compensating effect of harmonic filters will be improved. Besides, the value of the objective function will be accepted only if all harmonic individuals meet the limits. Otherwise, this set of data will be considered unqualified, and the objective value will be set to a positive value '1'.

III. GRASSHOPPER OPTIMIZATION ALGORITHM

GOA is a novel population-based heuristic algorithm proposed by Saremi *et al.* [14], which mimics the swarm behavior of grasshopper insects. The movement of a grasshopper is determined by three factors: social interaction, gravity force, and wind advection. The author established a mathematic model to simulate the swarming behavior of grasshopper

$$X_i = S_i + G_i + A_i \tag{28}$$

where X_i refers to the position of the $i-th$ grasshopper, S_i is the social interaction, G_i is the gravity force, and A_i is the wind advection.

The gravity force (G) is defined as follows:

$$G_i = -g\widehat{e_g} \tag{29}$$

where g is the gravitational constant and $\widehat{e_g}$ shows a unity vector towards the center of the earth.

The A component in Eq. 28 is calculated as follows:

$$A_i = -u\widehat{e_w} \tag{30}$$

where u is a constant drift and $\widehat{e_w}$ is a unit vector in the direction of wind.

Among all the three factors, the most important one is the social interaction that formulated as follows:

$$S_i = \sum_{j=1,j\neq i}^{N} s(d_{ij})\hat{d}_{ij} \tag{31}$$

where d_{ij} is the distance between *i-th* and *j-th* grasshopper that calculated as $d_{ij} = \left|x_j - x_i\right|$; $\hat{d}_{ij}$ is a unit vector from the *i-th* grasshopper to the *j-th* grasshopper, defined as $\hat{d}_{ij} = (x_j - x_i)/d_{ij}$. The s function reflects the social relationship strength in the grasshopper swarm. The s function is defined as follows:

$$s\left(r\right) = fe^{-\frac{r}{l}} - e^{-r} \tag{32}$$

where e is the natural logarithm, parameters f and l indicate the intensity of attraction and attractive length scale, respectively.

Substituting the S, G, and A in Eq.28, the equation can be expanded as follows:

$$X_i^d = \sum_{j=1,j\neq i}^{N} s\left(\left|x_j^d - x_i^d\right|\right)\frac{x_j - x_i}{d_{ij}} - g\widehat{e_g} + u\widehat{e_w} \tag{33}$$

However, this mathematical model cannot be used directly to solve optimization problems because the grasshoppers quickly reach the comfort zone and cannot converge to the target. Therefore, the author formulated another updating equation defined as Eq. 34:

$$X_i^d = c\left(\sum\nolimits_{j=1,j\neq i}^{N} c\frac{ub_d - lb_d}{2} s\left(\left|x_j^d - x_i^d\right|\right)\frac{x_j - x_i}{d_{ij}}\right) + \widehat{T_d} \tag{34}$$

where ub_d and lb_d represents the upper bound and lower bound of *d-th* dimension respectively, $\widehat{T_d}$ is the position of the best particle in the *d-th* dimension, the parameter c is linearly decreased during the iterations, which is calculated as follows:

$$c = cmax - l\frac{cmax - cmin}{L} \tag{35}$$

where $cmax$ is the maximum value, $cmin$ is the minimum value, L is the maximum number of iterations, and l indicates the current number of iterations.

The pseudo code of GOA is as follows:

Algorithm 1 GOA

```
/*Initialization*/
Initialize the swarm;
Initialize the parameters ($cmax$, $cmin$, $f$, $l$),
    population size ($Np$), maximum number ($L$);
Calculate the fitness of each search agent;
Find the best search agent $\hat{T_d}$;
while ($l < L$) do
    Update $c$ using Eq. 35;
    for 1: $Np$ do
        Normalize the distance between grasshoppers in
            [1], [4];
        Update the position of the current search agent ($X_i^d$)
            by the Eq. 34;
        Bring the current search agent back if it goes outside
            the boundaries;
    end for
    Update $\widehat{T_d}$ if there is a better solution;
    $l = l + 1$;
end while
Return $\widehat{T_d}$;
```

IV. IMPROVED GRASSHOPPER OPTIMIZATION ALGORITHM

A. NON-LINEAR COMFORT ZONE PARAMETER

Comfort zone parameter c in GOA is similar to the inertia weight ω in PSO [30], both of them decrease linearly during the iteration. Generally, a large parameter at the start of iteration keeps strong forces on global search while a small parameter in the late stage of iteration is conducive to a more precise local search. As the range of particles does not decrease linearly, the linearly decreased comfort zone cannot make full use of every iteration. Hence, we introduced a non-linear decreased parameter in this paper, which is defined as follows:

$$\omega(k) = \omega_{start} - (\omega_{start} - \omega_{end}) \left(\frac{k}{T_{max}}\right)^2 \tag{36}$$

where ω_{start} and ω_{end} are the parameters at the start and the end of iteration respectively, k refers to current iteration times, T_{max} is the maximum number of iterations. By Eq. 36, inertia weight ω is relatively large and gradually decreases during the early iterative process, improving the global search ability. In the later period, the ω decreases rapidly to apply strong local search for more precise solutions.

B. SCATTERED EXEMPLAR POOL MECHANISM

At the beginning of the GOA algorithm, the positions of grasshoppers are distributed randomly, and the target may locate in local best. Without a mechanism to jump out of local optimum and all grasshoppers moving towards the same target by social interaction forces, the swarm has poor global search ability and gets trapped in local optimum easily.

To avoid being trapped in local optima and enhance global search, we built an exemplar pool (*EP*) to replace the target ($\widehat{T_d}$) in GOA. The exemplar pool is composed of several particles with relatively high-quality fitness value. Inspired by reference [31], we take the distribution of particles in the exemplar pool into consideration to avoid the particles searching the same solution region repeatedly. The particles select their exemplars from *EP* randomly to explore solution space. As particles learn from multiple targets scattered in the solution space, the exploration ability is enhanced. Moreover, the *EP* is updated periodically to fully search the space around elite particles. Here, the periodic T is set to 5 based on the empirical evaluation. That is, the exemplar pool is updated in every five generations. The formula of exemplar pool (*EP*) is introduced from reference [31].

$$EP = \{S_1, S_2, \ldots, S_M | F(S_1) \geq F(S_2) \geq \ldots \geq F(S_M), \\ \text{for} \quad \forall S_i, S_{j \neq i}, \Delta(S_i, S_j \geq \gamma)\}$$

where M is the maximum number of candidate particles, F indicates the fitness value of selected exemplar, $\Delta(S_i, S_j)$ is the distance between two candidate solutions, γ is the minimum distance among candidate exemplars. To reduce the complexity, the parameter γ is acquired by empirical evaluation.

C. IMPROVED SOCIAL INTERACTION MECHANISM(ISI)

In GOA, the social interaction force determines how grasshopper move towards the target. The impact of social interaction is determined by function s, which divides the space between two grasshoppers into the repulsion region, comfort zone, and attraction region by the distance between them. When the distance is smaller than 2.079, the particle enters the repulsion region and moves away from another grasshopper. When larger than 2.079, the grasshopper is attracted by another one, which is defined as the attraction region. The grasshopper enters the comfort zone at the distance of 2.079 that social forces drop to zero. This distance mechanism enables grasshoppers to exploit or explore the space between two particles. The social interaction in Eq. 34 calculates the current interaction between grasshoppers and all the other grasshoppers in the swarm, affecting their movement towards the target. The accumulated distance effect cannot reflect whether to explore or exploit the space between particle and target and increase the complexity of calculation.

To tackle this problem, we modify the social interaction part in Eq. 34 as follows:

1) The grasshopper no longer calculates the distance between each particle in the whole swarm. To fully utilize the characteristic of function s, the current particle calculates the distance between its target and itself directly. If the distance is beyond the comfort zone, the grasshopper repels the target to explore the solution space. When it enters the attraction region, the particle moves around the target to exploit the solution space.

2) We proposed an adaptively changed comfort zone mechanism: At the beginning of the iteration, the particles should be uniform-distributed to search the space thoroughly. So we defined a comfort zone coefficient ε, which calculates as follows:

$$\varepsilon(i) = \frac{X_{imax} - X_{imin}}{Np} \tag{37}$$

where X_{imax}, X_{imin} is the maximum and minimum value of the position in *i-th* dimension respectively, *Np* is the number of search agents in the swarm.

In Eq. 37, we calculate the range of particles distributed in the *i-th* dimension and define the comfort zone parameter as range/*Np*. If the distance between two particles is larger than the comfort zone coefficient $\varepsilon(i)$, the particle will be repelled from another. Otherwise, it moves toward another. Meanwhile, the comfort zone coefficient ε updates periodically to adapt the change of positional range in the iteration.

3) In the original GOA, the author mapped the distance between two grasshoppers into [1, 4] to calculate the s function. In this paper, we map the distance into the interval [1,2.773], where 2.773 is the maximum point of s function. As Fig. 4(a) shows, x = 1.386 is the zero point of s function when its parameters subject to l = 2.0, f = 0.5. Therefore, we define x = 1.386 as the comfort zone. When the distance is smaller than 1.386, the particle enters the repulsion region. Conversely, attraction occurs in interval [1.386,2.773]. Particularly, as the distance in every dimension differs, we proposed a map method: First, we define the $\varepsilon(i)$ as the comfort zone of the *i-th* dimension. We map the distance interval [0, $\varepsilon(i)$] into [1,1.386]. Conversely, we map the distance into [1.386,2.773]

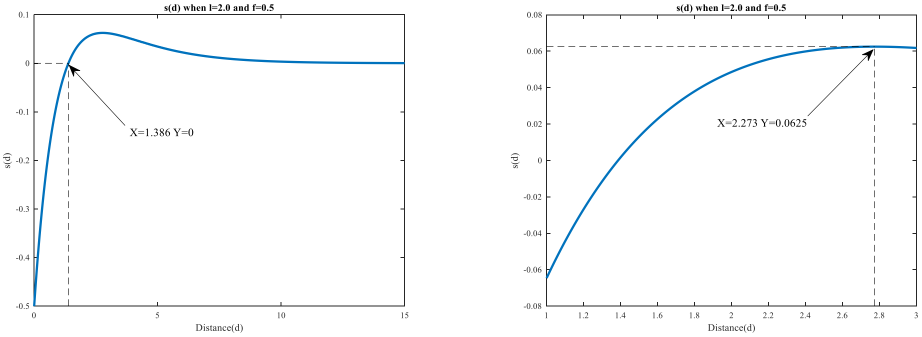


FIGURE 4. Function s when l = 2.0 and f = 0.5.

when larger than the $\varepsilon(i)$. The final mapping interval and s function are illustrated in Fig. 4, where we select the parameters of s function l = 2.0, f = 0.5.

The final formula of position updating is defined as follows:

$$X_{i_{new}}^{d} = \omega(k)\left(\frac{ub_d - lb_d}{2} s\left(\left|x_{EP(j)}^{d} - x_i^d\right|\right)\frac{x_{EP(j)}^{d} - x_i^d}{d_{ij}}\right) + x_{EP(j)}^{d} \quad (38)$$

where $x_{EP(j)}^{d}$ is the position of an elite particle randomly selected from exemplar pool (EP), x_i^d is the current position of i-th particle in d dimension, ub_d is the upper bound and lb_d is the lower bound of d-th dimension, $\omega(k)$ is the inertia weight mentioned before. Due to no distance calculation between all individuals, this strategy can effectively reduce the computational cost compared with the original social interaction in GOA.

D. TWO SUBPOPULATIONS WITH LEARNING STRATEGY

In the search process of GOA, the particles move around the target by the social interaction force, which enhances the exploitation ability around the global best. However, GOA has poor exploration ability and limited population diversity, for there is no global search mechanism in it. To overcome this flaw, we introduce and modify the learning strategy in ITLBO [22].

In ITLBO, the learners are divided into two groups by their fitness value: better learners and worse learners. The learning strategy is defined as follows:

$$X_{i_{new}}^{d} = \begin{cases} x_i + rand \cdot (x_j - x_k), iff\ (x_i) < f\ (x_{mean}) \\ x_i + rand \cdot (x_{r3} - x_{r4}) + rand \cdot (x_{r5} - x_{r6}), otherwise \end{cases} \quad (39)$$

where $j, k, i, r_3, r_4, r_5, r_6$ are random integers in $\{1, Np\}$, and $j \neq k \neq i, r_3 \neq r_4 \neq r_5 \neq r_6 \neq i$.

For the better leaners (i.e., $f\ (x_i) < f(x_{mean})$), they learn from the other two learners x_i and x_k. For the worse learners, they learn from the experience of four distinct learners, which enhances the global search ability and the diversity of the population.

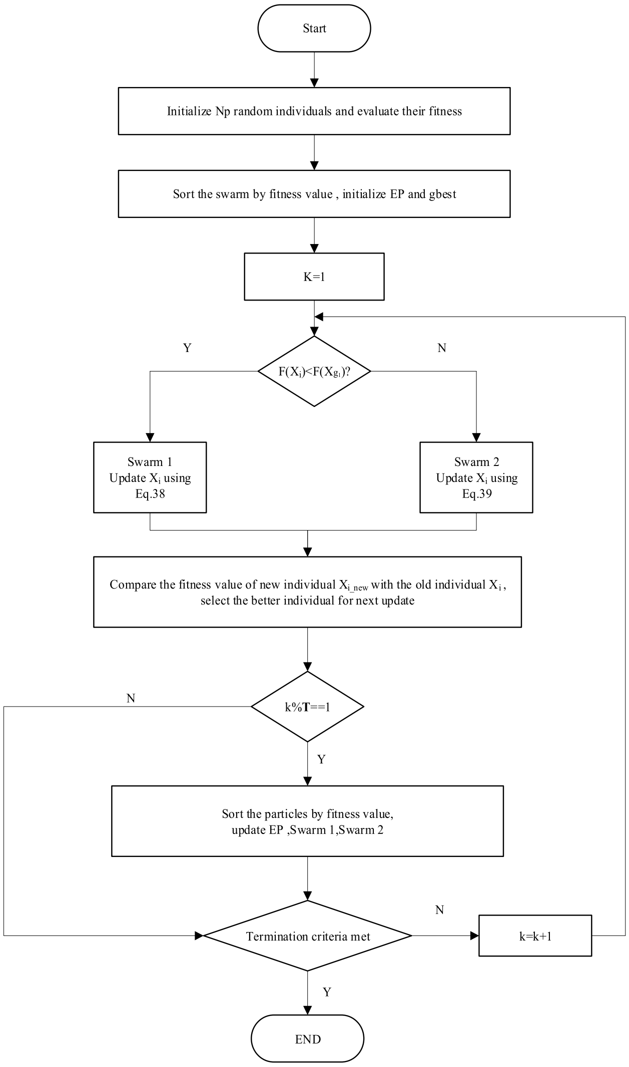

Start
Initialize Np random individuals and evaluate their fitness
Sort the swarm by fitness value , initialize EP and gbest
K=1
Y
N
F(X$_i$)<F(X$_{gi}$)?
Swarm 1
Update X$_i$ using
Eq.38
Swarm 2
Update X$_i$ using
Eq.39
Compare the fitness value of new individual X$_{i\_new}$ with the old individual X$_i$,
select the better individual for next update
N
k%T==1
Y
Sort the particles by fitness value,
update EP ,Swarm 1,Swarm 2
N
Termination criteria met
k=k+1
Y
END

FIGURE 5. The flowchart of IGOA.

Inspired by ITLBO, we also divide the whole swarm into two sub-swarms by the fitness value of particles. The particles are sorted firstly then the former g_1 number of particles are defined as better sub-swarm and the rest g_2 number of particles as worse sub-swarm. Different from ITLBO, we apply Eq. 38 to update position for better sub-swarms and the second equation in Eq. 39 for the worse sub-swarm. For the better sub-swarm, it moves around elite particles in *EP* to exploit. For the worse sub-swarm, Eq. 39 enables them to escape local optima and explore the search region, as well as improve population diversity.

E. PROCEDURE OF IGOA

The procedure of IGOA can be described with the pseudo code and flow chart.

Algorithm 2 IGOA

```
/*Initialization*/
Initialize the position $X_i$ of grasshoppers, $i \in \{1, 2 \cdots, N\}$;
Calculate the fitness value $F(x_i), i \in \{1, 2 \cdots, N\}$ using Eq. 27;
Sort the swarm by fitness value, initialize exemplar pool ($EP$);
while(condition of termination not met) do
    /*sub-swarm1*/
    for i = 1 to $g_1$ do
        Subswarm1= $[X_1, X_2, \ldots, X_{g_1}]$
        Update the position using Eq. 38;
    end for
    /*sub-swarm2*/
    for i= $g_1 + 1$ to $Np$ do
        Subswarm2= $[X_{g_1+1}, X_{g_1+2}, \ldots, X_{Np}]$
        Update the position using Eq. 39;
    end for
    Update the gbest;
    if update period T is reached
        Sort the swarm by fitness value;
        update sub-swarm1, sub-swarm2, $EP$;
    end if
end while
```

F. TIME COMPLEXITY ANALYSIS OF IGOA

The time complexity of the proposed IGOA relies on three parts: update the position of all search agents, evaluate the fitness of all agents, and sort search agents in population periodically. Suppose that the N, G, D, and T are the number of search agents, the number of generations, the dimensions of the parameters, and the period of updating exemplar pool, respectively. Since the quicksort algorithm is adopted, the time complexity is between $N\log N$ and N^2, corresponding to the best and worst case. Considering the worst case and the period of updating *EP*, the complexity of this part is GN^2/T. Therefore, the overall computational complexity of IGOA is:

$$\begin{aligned} O(IGOA) &= O(G*ND) + O(G*ND) + O(G*N^2/T) \\ &= O(GN\,(2D + N/T)) = O(GN^2) \end{aligned}$$

TABLE 1. Parameter settings of different algorithms.

Algorithm	Parameter setting
DE	Np=100, scale factor(F=0.5), Crossover Rate (CR=0.9)
LSHADE	Np=100, CR=0.5
ITLBO	Np=30
ABC	Number of onlooker bees =100, Np=100
GOA	Np=30, parameters of *s* function (l=2.0 f=0.5)
IGOA	Np=30, parameters of *s* function (l=2.0 f=0.5)
AGOA	Np=30, parameters of *s* function (l=1.5 f=0.5), c_{min}=0.1, c_{max}=0.9, w=5
OBLGOA	Np=30, OBL ratio=50%, parameters of *s* function (l=2.0 f=0.5)

Note: Np is the number of search agents

By contrast, the computational complexity of the original GOA is determined by two parts: update the position of all search agents and evaluate the fitness value of all agents. The overall computational complexity of GOA can be assessed as:

$$\begin{aligned} O\,(GOA) &= O\left(G*N^2D\right) + O\,(G*ND) \\ &= O\,(GND(N+1)) = O(GN^2D) \end{aligned}$$

Therefore, IGOA can significantly reduce the time complexity compared with the original GOA algorithm.

V. THE RESULTS AND ANALYSIS OF THE EXPERIMENT

To verify the performance of the IGOA, we applied it to identify the parameters of two HAPF configuration models. The numerical data of the model comes from three cases of an industrial plant in reference [29], where the total loads are 5100+4965j kVA with a low displacement power factor 0.7165. The system is supplied on a 4160 V line to line voltage and short-circuit capacity is approximately 80 MVA. Same as many previous studies[25], [32], the source and load harmonics are assumed to be time-invariant quantities; the load and source resistances are assumed to be frequency independent (R_{Ln} $=R_L$ and R_{Sn} $=R_S$). Both *VTHDlim* and *ITHDlim* are 5% based on the system under study. Table 2 illustrates the parameters of three cases of an industrial plant under study.

As discussed in Section II, we use the objective function (Eq. 27) to minimize the HP value. To indicate the superiority of the IGOA, we compared it with several well-established metaheuristic algorithms and other GOA variants. They are the standard GOA [14], artificial bee colony (ABC) [33], differential evolution (DE) [34], linear success history based adaptive differential evolution (L-SHADE) [13], improved teaching-learning-based optimization algorithm (ITLBO) [22], and two improved GOA algorithms: adaptive GOA (AGOA) [15], oppositional based learning GOA (OBLGOA) [19]. Note that OBLGOA is one of the latest grasshopper optimization algorithms proposed in 2018. For the above algorithms, their parameter settings are given in Table 1. These algorithms are applied to three cases for

TABLE 2. Case of an industrial plant [29].

Parameters and cases	Case 1	Case 2	Case 3
R_{S1} (Ω)	0.02163	0.02163	0.02163
X_{S1}(Ω)	0.2163	0.2163	0.2163
R_{L1}(Ω)	1.7421	1.7421	1.7421
X_{L1}(Ω)	1.696	1.696	1.696
V_{S1}(kV)	2.40	2.40	2.40
V_{S5}(%V_{S1})	0.00	2.00	4.00
V_{S7}(%V_{S1})	0.00	1.50	3.00
V_{S11}(%V_{S1})	0.00	1.00	2.00
V_{S13}(%V_{S1})	0.00	0.50	1.00
I_{L5}(%I_L)	40.00	40.00	40.00
I_{L7}(%I_L)	6.00	6.00	6.00
I_{L11}(%I_L)	2.00	2.00	2.00
I_{L13}(%I_L)	1.00	1.00	1.00

the HAPF configurations to calculate the statistical results of HP.

For a fair comparison, the maximum number of evaluations for the compared methods is set to 50000. All the algorithms were implemented in MATLAB R2016a and executed 31 independent runs. All comparative experiments were executed on a laptop PC with an Intel Core i5-7300HQ processor @ 2.50GHz, 8GB RAM.

A. RESULT ON HAPF CONFIGURATION 1

Table 3 shows the best, worst, mean, standard deviation value of HP and Success rate, as well as the total run time of each algorithm. We also conducted a Wilcoxon rank-sum test at 5% significant level to verify the significance of the results. The symbol "+1", "−1" denotes that IGOA is obviously better or worse than the compared algorithm. Symbol "0" denotes IGOA performs almost the same as the compared algorithm. To improve readability, the best of each HP value is highlighted in **boldface.** Additionally, the success rate refers to the times an algorithm obtains the best result in 31 runs. For unqualified solutions, they were excluded from the statistical result.

As can be seen from Table 3, the proposed IGOA achieves the best result among all the compared algorithms in terms of mean HP value and standard deviation value in case 1,2,3. Compared with GOA, IGOA performs significantly better in Mean and Min value, which demonstrates that IGOA can effectively escape the local optimum. For OBLGOA and AGOA, both of them achieved the second-best Min HP value in case 1, but their average performance is relatively low in all cases. Both DE and LSHADE provide highly competitive results in terms of min HP value and success rate in case 1 and 2. Despite the low success rate and worse mean value, ABC achieved the best Min HP value in case 2.

TABLE 3. Statistical result of HP value on the HAPF Config.1.

Case no.	Algorithm	Harmonic Pollution (in %)				Success rate (%)	Total CPU times(s)	Wilcoxon rank-sum test
		Min	Max	Mean	Std			
Case 1	IGOA	**0.236**	**0.236**	**0.236**	**$5.39*10^{-5}$**	**100.00**	158.4	
	DE	**0.236**	0.493	0.369	0.131	67.70	173.4	+1
	ABC	**0.236**	0.493	0.349	0.089	51.60	128.1	+1
	LSHADE	**0.236**	0.493	0.252	0.129	61.30	140.1	+1
	ITLBO	**0.236**	0.493	0.344	0.114	74.20	121.5	0
	GOA	0.244	0.725	0.470	0.229	3.32	637.7	+1
	AGOA	0.237	1.931	0.6491	0.363	12.90	582.8	+1
	OBLGOA	0.237	2.515	0.8989	0.6484	3.32	426.8	+1
Case 2	IGOA	2.751	2.753	**2.752**	**$4.820*10^{-4}$**	96.70	157.3	
	DE	2.752	2.952	2.778	0.068	87.10	174.3	0
	ABC	**2.741**	2.954	2.916	0.071	6.45	164.7	+1
	LSHADE	2.752	**2.752**	**2.752**	0.100	100.00	128.1	+1
	ITLBO	2.752	**2.752**	**2.752**	0.100	100.00	122.7	0
	GOA	2.752	4.292	2.997	0.297	3.32	624.6	+1
	AGOA	2.768	3.909	2.961	0.206	6.45	578.8	+1
	OBLGOA	2.856	4.482	3.179	0.385	6.45	432.0	+1
Case 3	IGOA	**5.672**	**5.888**	**5.725**	**0.090**	16.20	160.2	
	DE	**5.672**	**5.888**	5.839	0.092	25.80	168.3	+1
	ABC	5.895	5.925	5.901	0.030	6.45	171.2	+1
	LSHADE	**5.672**	5.923	5.861	0.075	12.90	140.2	0
	ITLBO	**5.672**	**5.888**	5.818	0.103	32.30	131.6	+1
	GOA	5.999	6.424	6.233	0.110	3.32	583.3	+1
	AGOA	5.936	6.529	6.161	0.183	3.32	593.8	+1
	OBLGOA	5.978	6.393	6.136	0.129	6.45	440.9	+1

TABLE 4. Optimized parameters of the best results for case studies with the HAPF Config.1.

Case no.	Algorithm	Optimized Parameters					
		$X_C(\Omega)$	$X_L(m\Omega)$	$G(\Omega)$	ITHD (%)	VTHD (%)	HP (%)
Case 1	**IGOA**	2.7093	103.59	20.00	0.200	0.125	**0.236**
	DE	2.7094	103.65	20.00	0.200	0.125	**0.236**
	ABC	2.7113	105.24	20.00	0.199	0.127	**0.236**
	LSHADE	2.7094	103.65	20.00	0.200	0.125	**0.236**
	ITLBO	2.7094	103.64	20.00	0.199	0.125	**0.236**
	GOA	2.7143	108.51	20.00	0.205	0.132	0.244
	FFSQP	2.7279	112.14	11.00	0.540	0.200	0.610
	AGOA	2.7095	106.07	20.00	0.128	0.199	0.237
	OBLGOA	2.7094	106.16	20.00	0.128	0.200	0.237
Case 2	**IGOA**	2.6995	91.47	20.00	0.512	2.703	2.751
	DE	2.6998	91.75	20.00	0.511	2.704	2.752
	ABC	2.6943	85.87	18.23	0.592	2.676	**2.741**
	LSHADE	2.6998	91.76	20.00	0.511	2.704	2.752
	ITLBO	2.6998	91.75	20.00	0.511	2.704	2.752
	GOA	2.6999	91.72	20.00	0.511	2.704	2.752
	FFSQP	2.7061	97.93	12.00	0.810	2.730	2.850
	AGOA	2.9698	97.45	20.00	0.564	2.711	2.768
	OBLGOA	2.7635	109.45	20.00	0.491	2.775	2.818
Case 3	**IGOA**	2.6161	$1.82*10^{-4}$	9.73	3.307	4.609	5.673
	DE	2.6159	$3.07*10^{-8}$	9.75	3.306	4.609	**5.672**
	ABC	5.2187	$3.92*10^{-3}$	16.80	3.507	4.738	5.744
	LSHADE	2.6159	$1.36*10^{-8}$	9.75	3.306	4.609	**5.672**
	ITLBO	2.6159	$3.88*10^{-7}$	9.75	3.306	4.609	**5.672**
	GOA	2.6537	37.01	6.39	3.847	4.603	5.999
	FFSQP	2.6171	$0.99*10^{-6}$	8.07	3.980	4.440	5.970
	AGOA	2.6921	$3.54*10^{-3}$	8.57	3.886	4.488	5.936
	OBLGOA	2.6836	$6.74*10^{-5}$	8.34	3.977	4.463	5.978

Concerning CPU run time, ITLBO achieves the best result in Min value in all cases with the shortest CPU run time. GOA has the highest CPU run time and among these algorithms, followed by AGOA and OBLGOA. By contrast, the CPU run time of IGOA is significantly less than other GOA algorithms, which further validates its improvement in computational complexity.

For HAPF Config.1, the optimized parameter and useful harmonic distortion value are presented in Table 4. The data is the best value among 31 runs for each algorithm. Due to the nonlinearity of the objective function, different optimized parameters can obtain almost the same HP value. We also compared the results with the previous study [12], which used the FFSQP method to optimize HAPF parameters. As can be seen in Table 4, the IGOA optimized three parameters of HAPF Config.1 and finally reduced the VTHD and ITHD value, thereby improved the compensating effect of the harmonic filters. Fig. 8 exhibits the value of compensated load voltage and supply current of individual harmonic contents. It shows that all results are well within IEEE standard limits [29].

Furthermore, Fig. 6 gives the box-plots, indicating the distribution of all the 31 independent runs results acquired by the eight different methods. Note that the sign "+" in the figure represents the abnormal value during the plotting. The span of the data distributions demonstrates that superior performance is also obtained by the IGOA algorithm. Fig. 7 gives the convergence curves of eight methods, which are plotted by using the average HP_{APP} value in 31 independent tests. Both DE and LSHADE converge faster than IGOA in case 2, but IGOA achieves the best results in all cases. For AGOA and OBLGOA, they outperform GOA in all cases but still always get trapped in local optima. As the No Free Lunch theory [35] indicates, every meta-heuristic algorithm has its characters, the improved GOA variants may not perform well in this problem. Despite IGOA has relatively low convergence speed, its mechanism to escape local optima is effective to achieve the best average value at the end.

Compared with seven other methods, the above analysis shows the effectiveness of IGOA in extracting parameters of HAPF Config.1, especially in terms of accuracy and reliability.

B. RESULT ON HAPF CONFIGURATION 2

Table 5 gives the statistical result of HP value on the HAPF Config.2. Obviously, the IGOA achieves the best Min, Mean, Std value of HP in all cases, which demonstrates its reliability and accuracy. By the Wilcoxon rank-sum test, we can see that IGOA outperforms other methods in case 1. Note that both DE and LSHADE provide very competitive results in case 2 and 3, for they received "0" during the Wilcoxon rank-sum test. Like the result in case 1, AGOA and OBLGOA

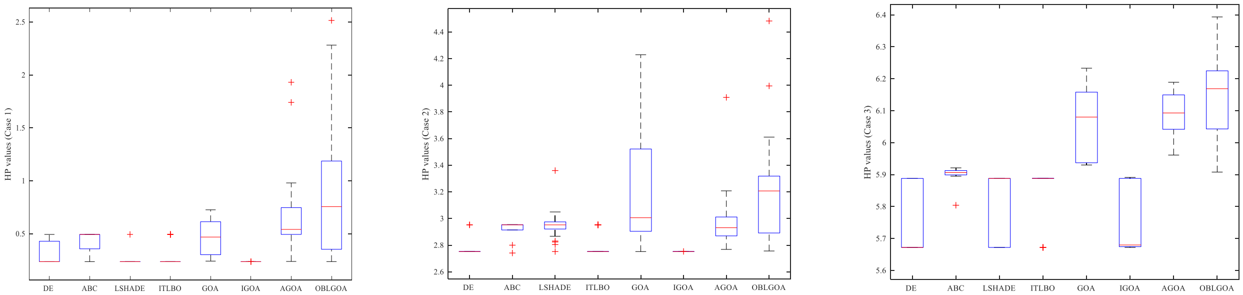


FIGURE 6. HP values box-plot in 31 runs of different algorithms for three cases of HAPF Config.1.

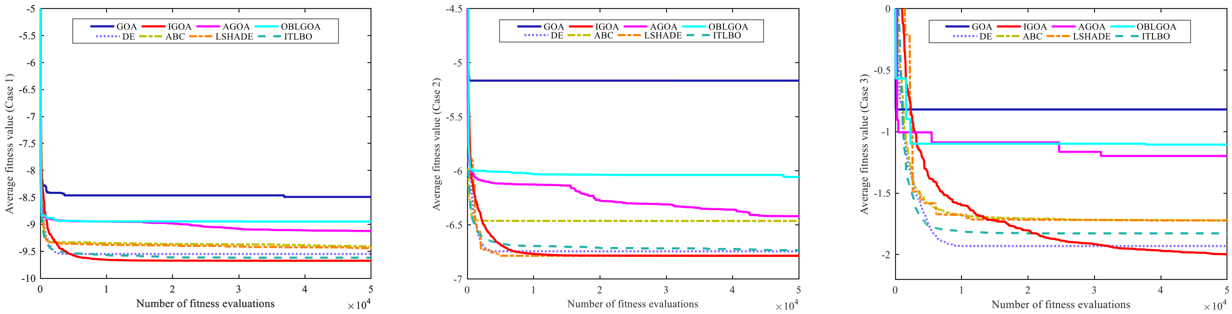
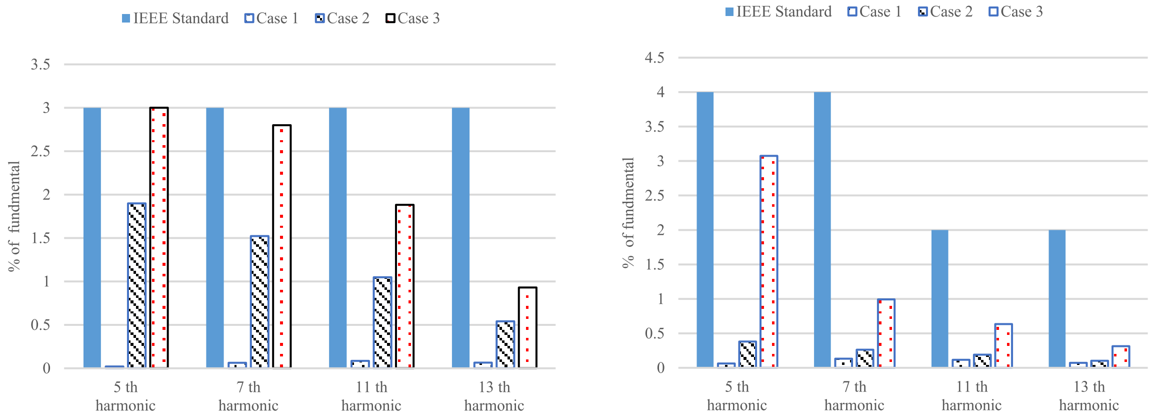


FIGURE 7. Convergence graphs of different algorithms for three cases of HAPF Config.1.



(a). Compensated individual harmonic of load voltage (V_{Ln})

(b). Compensated individual harmonic of supply current (I_{Sn})

FIGURE 8. Harmonic content for compensated HAPF Config.1.

can get very close to the best Min value in case 1 but remain poor on the average performance.

Box-plots and Convergence curves are given in Fig. 9 and Fig. 10. Similar to the HAPF Config.1, IGOA achieves very competitive results. Especially in case 3, the average value obtained by IGOA is significantly better than other methods. The box-plot also shows that the distribution of the result is smaller than others, which further proves that IGOA can get a more reliable result.

From Fig. 11 and Table 6, it can be observed that the results (ITHD, VTHD, HP) of Config.2 in three cases are highly similar to those in Config.1. A close review of the equation in Section II can account for this. From Eq. 6 and Eq. 10, we can see that the value of compensated current mainly depends on inductive reactance, a subtle change in other parameters plays an insignificant role in supply current harmonics. Nevertheless, a subtle difference still has important research significance.

C. EFFECTS OF NEW STRATEGIES IN IGOA

To verify the effect of the proposed multiple strategies, we also developed three IGOA variants, namely IGOA-1, IGOA-2, IGOA-3, and IGOA-4. To be specific, IGOA-1 replaces the social interaction of the original GOA with

TABLE 5. Statistical result of HP value on the HAPF Config.2.

Case no.	Algorithm	Harmonic Pollution (in %)				Success rate (%)	Total CPU times(s)	Wilcoxon rank-sum test
		Min	Max	Mean	Std			
Case 1	IGOA	**0.227**	**0.227**	**0.227**	**$3.38*10^{-5}$**	100	143.6	
	DE	**0.227**	0.458	0.309	0.113	64.52	169.3	+1
	ABC	0.231	0.459	0.434	0.067	3.23	136.8	+1
	LSHADE	**0.227**	0.459	0.265	0.087	83.87	135.5	+1
	ITLBO	**0.227**	0.459	0.272	0.093	80.64	115.3	+1
	GOA	0.239	0.726	0.451	0.201	3.23	604.3	+1
	AGOA	**0.227**	2.455	0.526	0.408	3.23	595.4	+1
	OBLGOA	0.229	2.175	0.884	0.550	3.23	419.4	+1
Case 2	IGOA	**2.753**	**2.756**	**2.754**	**$6.23*10^{-4}$**	6.45	145.4	
	DE	2.754	2.949	2.786	0.073	83.87	139.5	0
	ABC	2.951	3.452	3.055	0.979	3.23	130.1	+1
	LSHADE	2.754	2.95	2.767	0.049	93.54	125.6	0
	ITLBO	2.754	2.949	2.798	0.083	77.42	115.4	0
	GOA	2.808	4.187	3.094	0.422	3.23	568.1	+1
	AGOA	2.807	4.373	3.122	0.369	3.23	597.6	+1
	OBLGOA	2.844	4.645	3.159	0.505	3.23	428.1	+1
Case 3	IGOA	**5.681**	**5.906**	**5.729**	**0.091**	29.03	117.8	
	DE	**5.681**	**5.906**	5.797	0.114	29.03	169.3	0
	ABC	5.915	5.933	5.924	0.005	3.23	135.7	+1
	LSHADE	**5.681**	5.943	5.835	0.107	29.03	145.7	0
	ITLBO	**5.681**	5.906	5.847	0.100	25.81	111.6	+1
	GOA	5.901	6.979	6.321	0.341	3.23	573.6	+1
	AGOA	5.893	6.925	6.349	0.334	6.45	578.1	+1
	OBLGOA	5.902	7.039	6.353	0.402	3.23	403.1	+1

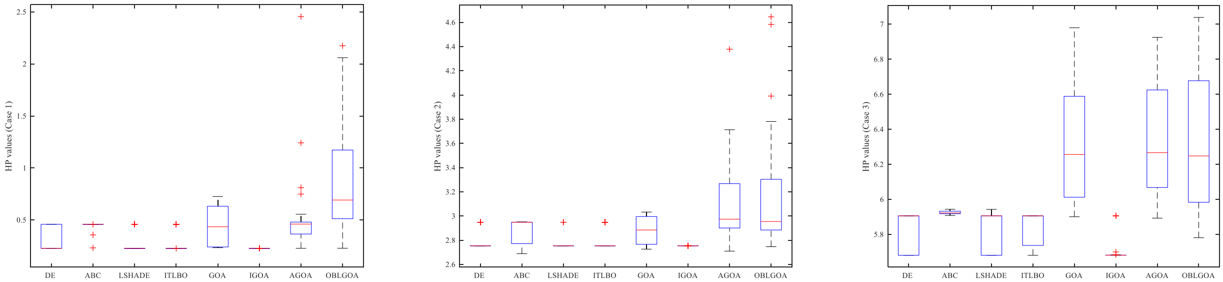


FIGURE 9. HP values box-plot in 31 runs of different algorithms for three cases of HAPF Config.2.

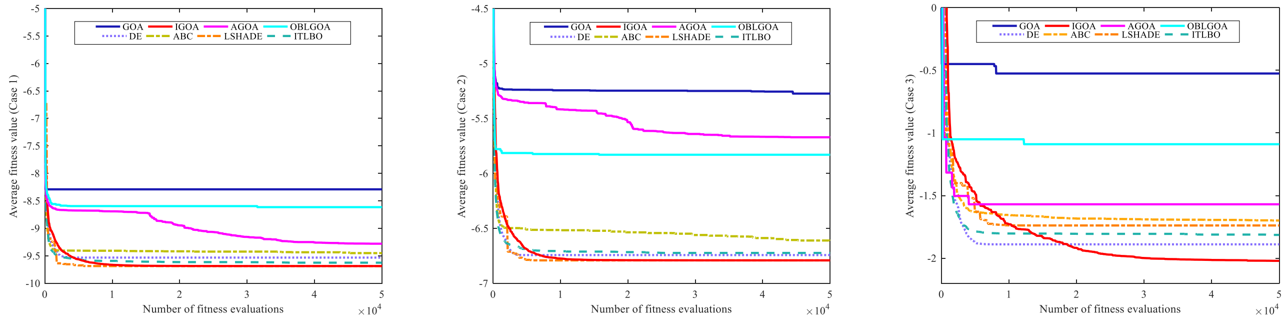
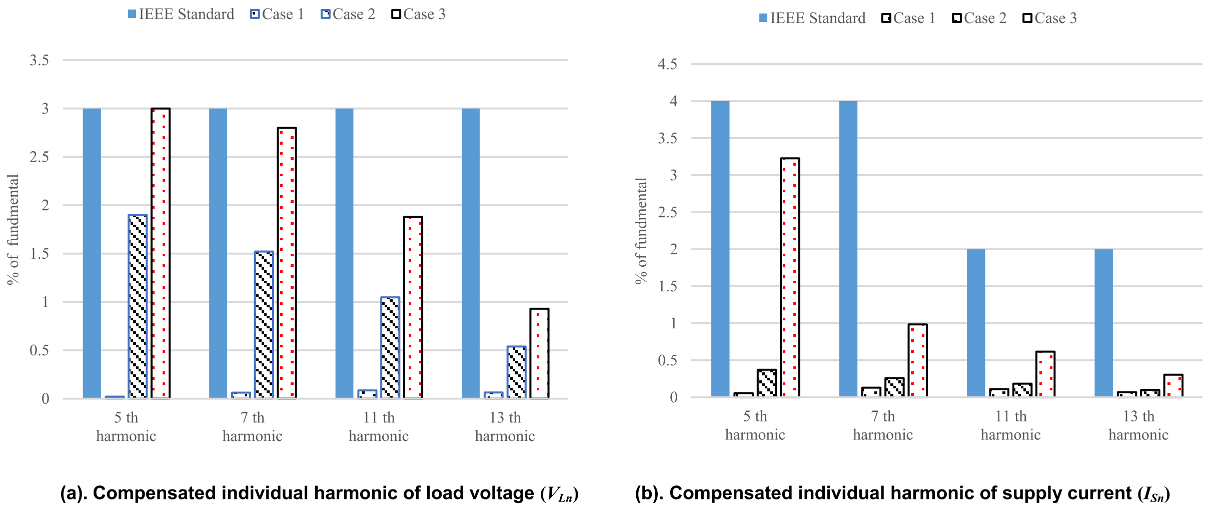


FIGURE 10. Convergence graphs of different algorithms for three cases of HAPF Config.2.

the improved social interaction (ISI) strategy. However, the learning target in ISI is the best particle instead of elite particles in the EP. IGOA-2 introduces the exemplar pool (EP) strategy into the original GOA. IGOA-3 introduces two sub-swarm with learning (TSL) strategy into the original GOA. For IGOA-4, we integrated the ISI and EP strategy into the



(a). Compensated individual harmonic of load voltage (V_{Ln})

(b). Compensated individual harmonic of supply current (I_{Sn})

FIGURE 11. Harmonic content for compensated HAPF Config.2.

TABLE 6. Optimized parameters of the best results for case studies with the HAPF Config.2.

Case no.	Algorithm	Optimized Parameters					
		$X_C(\Omega)$	$X_L(m\Omega)$	$G(\Omega)$	ITHD (%)	VTHD (%)	HP (%)
Case 1	IGOA	2.7099	104.20	20.00	0.193	0.120	**0.227**
	DE	2.7099	104.16	20.00	0.193	0.120	**0.227**
	ABC	2.7095	103.71	18.98	0.203	0.127	0.240
	LSHADE	2.7099	104.16	20.00	0.193	0.120	**0.227**
	ITLBO	2.7103	104.17	20.00	0.193	0.120	**0.227**
	GOA	2.7140	108.52	20.00	0.197	0.126	0.234
	AGOA	2.7101	103.80	20.00	0.193	0.120	**0.227**
	OBLGOA	2.7121	106.50	20.00	0.193	0.123	0.229
Case 2	IGOA	2.7013	93.56	20.00	0.500	2.709	**2.754**
	DE	2.7013	93.31	20.00	0.500	2.708	**2.754**
	ABC	5.4511	199.78	20.00	0.624	2.884	2.951
	LSHADE	2.7013	93.31	20.00	0.500	2.708	**2.754**
	ITLBO	2.7013	93.31	20.00	0.500	2.708	**2.754**
	GOA	2.7154	107.77	20.00	0.476	2.768	2.808
	AGOA	2.7153	107.50	20.00	0.475	2.766	2.807
	OBLGOA	2.7201	113.52	18.66	0.522	2.796	2.844
Case 3	IGOA	2.6160	$1.1*10^{-2}$	10.29	3.313	4.616	**5.681**
	DE	2.6160	$2.34*10^{-5}$	10.31	3.312	4.615	**5.681**
	ABC	5.2193	3.92	18.84	3.524	4.748	5.912
	LSHADE	2.6161	$9.39*10^{-8}$	10.31	3.312	4.615	**5.681**
	ITLBO	2.6160	$2.16*10^{-9}$	10.31	3.312	4.615	**5.681**
	GOA	5.2200	3.36	18.66	3.567	4.738	5.930
	AGOA	2.6205	$7.55*10^{-3}$	8.96	3.812	4.498	5.895
	OBLGOA	2.6224	$1.52*10^{-3}$	8.932	3.828	4.494	5.902

original GOA. By comparing the IGOA and its variants, we can determine the performance of the ISI, EP, TSL, and ISI+EP strategy of the proposed IGOA.

For the above methods, we use the Mean HP value in all cases to compare the average performance of them. As can be seen in Table 7, IGOA-4 gets a very close result to IGOA in all cases, followed by IGOA-3. For the single strategy, the TSL strategy outperforms others in all cases. For the EP and ISI, the single strategy integrated to the original GOA can make slight improvement but remain weak. However, the combination of the two strategies effectively works and achieves satisfactory results in all cases.

Fig. 12 also gives the convergence curves of the IGOA variants and the original GOA. Due to the similarity between the two HAPF configurations and space limits, we only plot it in HAPF Config.1. For IGOA-1, IGOA-2, and GOA, they converge very fast in the early iteration process but always get trapped in local optima. For IGOA-4, it is worth mentioning

TABLE 7. Statistical result of Mean HP value for IGOA variants.

Algorithm	HP value	HAPF Configuration 1			HAPF Configuration 2		
		Case 1	Case 2	Case 3	Case 1	Case 2	Case 3
GOA	Mean	0.470	2.997	6.233	0.451	3.094	6.321
IGOA-1	Mean	0.546	3.565	6.408	0.561	3.481	6.049
IGOA-2	Mean	0.577	3.516	6.194	0.322	3.482	6.404
IGOA-3	Mean	0.351	2.851	5.900	0.309	2.792	5.864
IGOA-4	Mean	0.250	2.758	5.765	0.235	2.756	5.806
IGOA	Mean	0.236	2.752	5.725	0.227	2.754	5.729

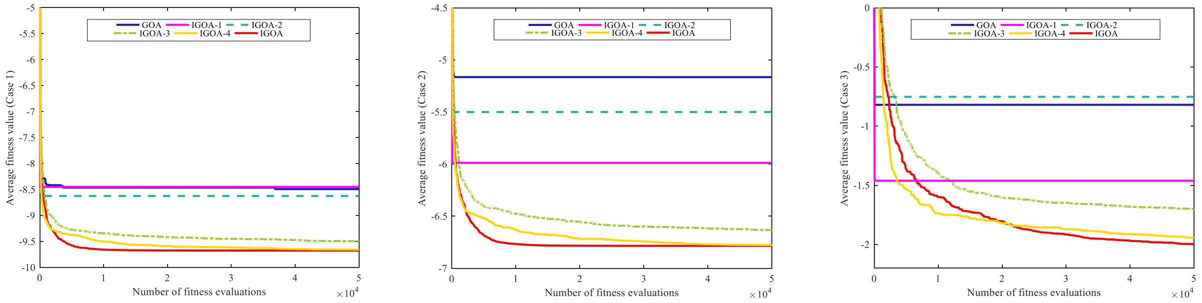


FIGURE 12. Convergence graphs of IGOA variants for three cases of HAPF Config.1.

that it achieved the same results with IGOA in case 1 and 2, but it converges slower at the early iteration process. In case 3, it converges faster than IGOA and gets a very close result to IGOA at the end. Compared with IGOA-1 and IGOA-2, it can effectively escape local optima during the iteration process. Compared with IGOA-3, it converges faster and achieves better results in all cases. The above analysis demonstrates that the combination of EP and ISI strategy plays a key role in IGOA's performance.

VI. CONCLUSION

This paper proposed an improved GOA(IGOA) algorithm for estimating the parameters of HAPF models. In IGOA, the exemplar pool is first introduced to replace the target in the original GOA for enhancing the searching ability and escaping local optima. We also proposed the improved social interaction (ISI) strategy, where grasshoppers get attracted or repulsed from an elite particle in the exemplar pool to balance exploration and exploitation ability. Since the ISI strategy does not compute the distance between different individuals, it can significantly decrease the computational cost of the method. Moreover, we divided the swarm into two subswarms and introduced a learning strategy. Finally, the drawback of IGOA should be noticed. The convergence speed of IGOA is relatively slower than other state-of-art meta-heuristic algorithms. Although it can get the best results in 31 runs, its total CPU runtime is not optimal among all the compared algorithms. In conclusion, IGOA can provide the most reliable overall performance on parameter extraction of HAPF models among all the compared methods, thereby significantly improve the compensating effect of HAPF filters.

In the future works, IGOA will be extended to solve the discrete optimization problems and multi-objective optimization problems, such as feature selection, image segmentation, and renewable energy problems. Applying the IGOA to optimize the photovoltaic cells system will also be an interesting research direction. Furthermore, some other improvements will be proposed to increase the convergence speed of IGOA further.

REFERENCES

[1] B.-R. Lin, B.-R. Yang, and H.-R. Tsai, "Analysis and operation of hybrid active filter for harmonic elimination," *Electr. Power Syst. Res.*, vol. 62, no. 3, pp. 191–200, Jul. 2002.

[2] Z. Shuai, A. Luo, W. Zhu, R. Fan, and K. Zhou, "Study on a novel hybrid active power filter applied to a high-voltage grid," *IEEE Trans. Power Del.*, vol. 24, no. 4, pp. 2344–2352, Oct. 2009.

[3] A. Luo, Z. Shuai, W. Zhu, Z. J. Shen, and C. Tu, "Design and application of a hybrid active power filter with injection circuit," *IET Power Electron.*, vol. 3, no. 1, pp. 54–64, Jan. 2010.

[4] Y.-P. Chang and C.-J. Wu, "Optimal multiobjective planning of large-scale passive harmonic filters using hybrid differential evolution method considering parameter and loading uncertainty," *IEEE Trans. Power Del.*, vol. 20, no. 1, pp. 408–416, Jan. 2005.

[5] Y.-M. Chen, "Passive filter design using genetic algorithms," *IEEE Trans. Ind. Electron.*, vol. 50, no. 1, pp. 202–207, Feb. 2003.

[6] S. G. Zhao, Y. P. Wang, and L. C. Jiao, "Adaptive genetic algorithm based optimal design approach for passive power filters," *Proc.-Chin. Soc. Elect. Eng.*, vol. 24, no. 7, pp. 173–176, 2004.

[7] T. Chunming, L. An, and L. Juan, "Multi-objective optimal design of passive power filters," *Proc.-Chin. Soc. Elect. Eng.*, vol. 22, no. 3, pp. 17–21, 2002.

[8] A. F. Zobaa, "Mixed-integer distributed ant colony multi-objective optimization of single-tuned passive harmonic filter parameters," *IEEE Access*, vol. 7, pp. 44862–44870, 2019.

[9] M. Mohammadi, "Bacterial foraging optimization and adaptive version for economically optimum sitting, sizing and harmonic tuning orders setting of LC harmonic passive power filters in radial distribution systems with linear and nonlinear loads," *Appl. Soft Comput.*, vol. 29, pp. 345–356, Apr. 2015.

[10] S. H. E. A. Aleem, A. F. Zobaa, M. E. Balci, and S. M. Ismael, "Harmonic overloading minimization of frequency-dependent components in harmonics polluted distribution systems using Harris hawks optimization algorithm," *IEEE Access*, vol. 7, pp. 100824–100837, 2019.

[11] N. He, D. Xu, and L. Huang, "The application of particle swarm optimization to passive and hybrid active power filter design," *IEEE Trans. Ind. Electron.*, vol. 56, no. 8, pp. 2841–2851, Aug. 2009.

[12] A. F. Zobaa, "Optimal multiobjective design of hybrid active power filters considering a distorted environment," *IEEE Trans. Ind. Electron.*, vol. 61, no. 1, pp. 107–114, Jan. 2014.

[13] P. P. Biswas, P. N. Suganthan, and G. A. J. Amaratunga, "Minimizing harmonic distortion in power system with optimal design of hybrid active power filter using differential evolution," *Appl. Soft Comput.*, vol. 61, pp. 486–496, Dec. 2017.

[14] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: Theory and application," *Adv. Eng. Softw.*, vol. 105, pp. 30–47, Mar. 2017.

[15] J. Wu, H. Wang, N. Li, P. Yao, Y. Huang, Z. Su, and Y. Yu, "Distributed trajectory optimization for multiple solar-powered UAVs target tracking in urban environment by adaptive grasshopper optimization algorithm," *Aerosp. Sci. Technol.*, vol. 70, pp. 497–510, Nov. 2017.

[16] J. Luo, H. Chen, Q. Zhang, Y. Xu, H. Huang, and X. Zhao, "An improved grasshopper optimization algorithm with application to financial stress prediction," *Appl. Math. Model.*, vol. 64, pp. 654–668, Dec. 2018.

[17] Z. Elmi and M. O. Efe, "Multi-objective grasshopper optimization algorithm for robot path planning in static environments," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Feb. 2018, pp. 244–249.

[18] A. A. El-Fergany "Electrical characterisation of proton exchange membrane fuel cells stack using grasshopper optimiser," *IET Renew. Power Gener.*, vol. 12, no. 1, pp. 9–17, 2017.

[19] A. A. Ewees, M. A. Elaziz, and E. H. Houssein, "Improved grasshopper optimization algorithm using opposition-based learning," *Expert Syst. Appl.*, vol. 112, pp. 156–172, Dec. 2018.

[20] S. Arora and P. Anand, "Chaotic grasshopper optimization algorithm for global optimization," *Neural Comput. Appl.*, vol. 31, no. 8, pp. 4385–4405, Aug. 2019.

[21] N. Lynn and P. N. Suganthan, "Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation," *Swarm Evol. Comput.*, vol. 24, pp. 11–24, Oct. 2015.

[22] S. Li, W. Gong, X. Yan, C. Hu, D. Bai, L. Wang, and L. Gao, "Parameter extraction of photovoltaic models using an improved teaching-learning-based optimization," *Energy Convers. Manage.*, vol. 186, pp. 293–305, Apr. 2019.

[23] F. Z. Peng, H. Akagi, and A. Nabae, "A novel harmonic power filter," in *Proc. 19th Annu. IEEE Power Electron. Spec. Conf. (PESC)*, Apr. 1988, pp. 1151–1159.

[24] F. Z. Peng, H. Akagi, and A. Nabae, "A new approach to harmonic compensation in power systems—A combined system of shunt passive and series active filters," *IEEE Trans. Ind. Appl.*, vol. 26, no. 6, pp. 983–990, 1990.

[25] S. H. E. A. Aleem, A. F. Zobaa, and M. M. A. Aziz, "Optimal C-type passive filter based on minimization of the voltage harmonic distortion for nonlinear loads," *IEEE Trans. Ind. Electron.*, vol. 59, no. 1, pp. 281–289, Jan. 2012.

[26] S. Rahmani, A. Hamadi, and K. Al-Haddad, "A comprehensive analysis of hybrid active power filter for power quality enhancement," in *Proc. 38th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2012, pp. 6258–6267.

[27] H. Fujita and H. Akagi, "A practical approach to harmonic compensation in power systems-series connection of passive and active filters," *IEEE Trans. Ind. Appl.*, vol. 27, no. 6, pp. 1020–1025, Nov./Dec. 1991.

[28] R. Langella and A. Testa, *IEEE Standard Definitions for the Measurement of Electric Power Quantities Under Sinusoidal, Nonsinusoidal, Balanced, or Unbalanced Conditions*, IEEE Standard 1459-2010, 2010.

[29] *IEEE Recommended Practices and Requirements for Harmonic Control in Electrical Power Systems*, IEEE Standard 519-2014, F II, I, New York, NY, USA, 1993, p. 1.

[30] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, Nov./Dec. 1995, pp. 1942–1948.

[31] Z. Ren, A. Zhang, C. Wen, and Z. Feng, "A scatter learning particle swarm optimization algorithm for multimodal problems," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1127–1140, Jul. 2014.

[32] A. F. Zobaa, "A new approach for voltage harmonic distortion minimization," *Electr. Power Syst. Res.*, vol. 70, no. 3, pp. 253–260, Aug. 2004.

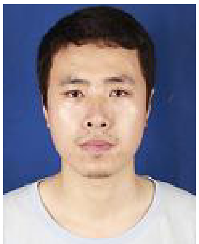
[33] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Fac. Eng., Dept. Comput. Eng., Erciyes Univ., Kayseri, Turkey, Tech. Rep. tr06, 2005, pp. 1–10, vol. 200.

[34] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[35] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

JUNRU HUANG is currently pursuing the B.Sc. degree with the School of Information Engineering, Nanchang University, Nanchang, China.

His current research interests include particle swarm optimization, genetic algorithms, and other computational intelligence techniques.



CHUNQUAN LI (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees from Nanchang University, Nanchang, China, in 2002, 2007, and 2015, respectively.

He has been with the School of Information Engineering, Nanchang University, since 2002, where he is currently a Professor and a Young Scholar of Ganjiang River. He is also a Visiting Professor with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada. He has published over 30 research articles. His current interests include computing intelligence, haptics, virtual surgery simulation, robotics, and their applications to biomedical engineering.

ZHILING CUI is currently pursuing the B.Sc. degree with the School of Information Engineering, Nanchang University, Nanchang, China.

His current research interests include teaching-learning-based optimization, moth-flame optimization, and other computational intelligence techniques.

LEYINGYUE ZHANG is currently pursuing the B.Sc. degree with the School of Information Engineering, Nanchang University, Nanchang, China.

Her current research interests include particle swarm optimization, moth-flame optimization, and other computational intelligence techniques.

WANXUAN DAI is currently pursuing the degree with the School of Information Engineering, Nanchang University, Nanchang, Jiangxi, China.

His current research interests include swarm particle swarm optimization, genetic algorithms, and other computational intelligence techniques.