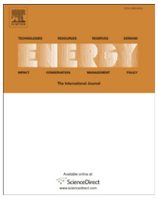
Energy 251 (2022) 123863

Contents lists available at ScienceDirect

Energy

journal homepage: www.elsevier.com/locate/energy



Maximum power point tracking and parameter estimation for multiple-photovoltaic arrays based on enhanced pigeon-inspired optimization with Taguchi method

Jeng-Shyang Pan [a, d], Ai-Qing Tian [a], Václav Snášel [b], Lingping Kong [b], Shu-Chuan Chu [a, c, *]

[a] College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, 266590, China
[b] Faculty of Electrical Engineering and Computer Science, VŠB-Technical University of Ostrava, Ostrava, Czech Republic
[c] College of Science and Engineering, Flinders University, 1284 South Road, Tonsley, SA, 5042, Australia
[d] Department of Information Management, Chaoyang University of Technology, Taichung, 413310, Taiwan

ARTICLE INFO

Article history:
Received 7 January 2022
Received in revised form
11 February 2022
Accepted 28 March 2022
Available online 4 April 2022

Keywords:
Pigeon-inspired optimization
Maximum power point tracking
Partially shaded conditions
Parameters estimation
Photovoltaic (PV) system
Taguchi method

ABSTRACT

The simulation, control and optimization of photovoltaic (PV) modules require the extraction of parameters from actual data and the construction of highly accurate PV cells. Multiple PV modules supplying power to a common load is the most common form of power distribution in PV systems. In these PV systems, providing separate maximum power point tracking (MPPT) technology for each PV module would increase the cost of the entire system. Determining how to accurately identify the internal parameter information of the PV modules and control the MPPT technology is the problem solved in this paper. we proposes an improved pigeon-inspired optimization (PIO) algorithm based on Taguchi method to solve the above problems. In this paper, we use the CEC2014 test library for testing and cross-sectional comparison. Experimental results show that the PIO algorithm based on Taguchi method is more competitive than other algorithms. The proposed algorithm uses measurement data to extract the unknown parameter in the PV modules and then uses this information to optimize the MPPT of all PV systems under partially shaded conditions (PSCs). Simulation results demonstrate the fitness value of the unknown parameters extracted by TPIO is 9.7525 × 10^{-4}, which is better than the compared algorithms.

1. Introduction

With the continuous development and progress of human society, the excessive use of fossil fuels has caused increasingly serious problems, such as environmental pollution and resource shortages [40,44]. The development of new energy technologies to replace fossil fuels is an urgent task at present [7,76]. The issue of use energy has become a concern for researchers in the context of economic development [32,63]. Among the various renewable energy sources available, such as wind, water [69], biomass and hydrogen, the most commonly used to generate electricity is solar energy [24,54,62]. In recent years, the development of solar photovoltaic (PV) systems based on solar energy has been very promising, because it is sustainable, clean and convenient [10,71,77]. PV systems can convert solar energy into electricity. Unknown issues within PV systems often arise due to internal causes and exposure to the outdoor environment, which is constantly changing. The actual performance of PV systems depends on unknown internal factors, which can change when the external conditions change. Therefore, an efficient and reliable parameter extraction method should be used to estimate the values of unknown parameters inside systems according to the available measurement data [14,79].

The power generated by a given PV system is mainly dependent on the radiation intensity of the sun, and the temperature in the installation environment [72]. Since these direct influences are constantly changing over time, it is necessary to develop a control logic that continuously monitors the current and voltage at both ends of systems. In addition, the control signals must be updated accordingly. Furthermore, to ensure the optimal operation of PV systems, it is necessary to have the voltage at both ends equal to the level of the corresponding maximum power. Various conventional

* Corresponding author. College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, 266590, China.

E-mail addresses: jengshyangpan@gmail.com (J.-S. Pan), stones12138@163.com (A.-Q. Tian), vaclav.snasel@vsb.cz (V. Snášel), konglingping2007@163.com (L. Kong), scchu0803@gmail.com (S.-C. Chu).

https://doi.org/10.1016/j.energy.2022.123863
0360-5442/

Nomenclature

I_s	saturation current
I_{mp}	maximum power current
I_{ph}	photocurrent in photovoltaic modules
I_{sc}	short-circuit current
R_p	shunt resistance
V_{mp}	maximum power voltage
V_{oc}	open-circuit voltage
ABC	artificial bee colony
ACO	ant colony optimization
CSO	cat swarm optimization
DDM	double-diode model
DPSO	discrete particle swarm optimization
FMO	fish migration optimization
GP	global peak
HC	hill climbing method
HSP	harmonic search process
IALO	improved ant lion optimizer
IHC	incremental hill climbing method
LP	local peaks
MPPT	maximum power point tracking
P—V	power-voltage
PIO	pigeon-inspired optimization
PSCs	partially shaded conditions
PSO	particle swarm optimization
PV	photovoltaic modules
QUATRE	QUasi-Affine TRansformation evolutionary
SCA	sine-cosine algorithm
SDM	single-diode model
SI	Swarm intelligence
STNEP	static transmission network expansion planning
TPIO	pigeon-inspired optimization algorithm based on Taguchi method
V—I	volt-ampere
WOA	whale optimization algorithm

single-stage maximum power point tracking (MPPT) algorithms have been proposed [2,18,19,27,33,73], and used to extract the maximum power value from PV systems under different operating conditions. In the literature [47,48,58], PV systems are implemented using MPPT based on Fibonacci searches, by comparing the measured power values of two operating points and then decides the next direction of movement of the operating point after making a judgment. The Fibonacci search approach and the hill climbing (HC) method yield similar performance, the only difference between them is that the step size is determined by the Fibonacci series in the Fibonacci approach. Therefore, both the traditional HC algorithm and the Fibonacci search method are suitable for searching local maximum power value points under specific conditions.

If a PV system is partially obscured by clouds, trees or buildings, it is difficult for the conventional MPPT algorithm to guarantee the maximum power output of the whole system. If a PV system is a parallel or series connection of modules with different optimal currents due to uneven insolation, the power-voltage (P—V) characteristic curve will often have an MPP. Notably the optimal current for each PV systems can be considered to be proportional to the sunlight intensity. Under these conditions, the traditional MPPT algorithm will most likely fall to a local optimum, the final value will be found near the local maximum power value point, and the global maximum power value point will not be reached. Therefore, the efficiency of PV systems under covered conditions is often poor, and improvements are needed.

Swarm intelligence (SI) algorithms have been proven effective in many applications [12,55,68] and include the particle swarm optimization (PSO) [34], ant colony optimization (ACO) [15], artificial bee colony (ABC) [31], fish migration optimization (FMO) [57,59], QUasi-Affine TRansformation evolutionary (QUATRE) [29,43], and cat swarm optimization (CSO) [13,67,81]. These algorithms can be applied to solve PV system problems. Researchers have proposed various methods to obtain the specific parameters for PV modules, and these methods mainly fall into four categories: analytical methods, direct methods, numerical calculations and other types of traditional methods [78]. However, the use of these types of optimization methods is highly limited by various constraints [49]. The SI algorithm has become very popular in recent years, that can find an optimal solution in a satisfactory time by continuously iterating over the search range [61]. Niu et al. [52] proposed a biogeography-based algorithm for variation strategy optimization. Shayeghi et al. [66] applied the discrete particle swarm optimization (DPSO) algorithm to the static transmission network expansion planning (STNEP) problem to prevent premature convergence, which largely improved the stability of the network with almost the same expansion cost. A parameter estimation method based on a harmonic search process (HSP) was proposed by Askarzadeh et al. [5]. Awadallah [8] used five different bacterial foraging algorithms to extract the unknown internal parameters of the single-diode model (SDM) and a double-diode model (DDM). Wu et al. [74] proposed an improved ant lion optimizer (IALO) to improve the accuracy of solar cell model parameter identification. Oliva et al. [56] proposed an ABC algorithm to solve the problem of parameter identification. Kang et al. [30] proposed an improved cuckoo search algorithm that combines an opposition learning scheme, a dynamic adaptation state and a Levi flight phase. Merchaoui et al. [45] and Nunes et al. [53] used an adaptive mutation strategy and guaranteed convergence to improve the traditional particle swarm algorithm and determine the accuracy of photovoltaic parameter measurements with different photovoltaic models considering the influence of different noises. Aydin et al. [9] combined the basic whale optimization algorithm (WOA) with the sine-cosine algorithm (SCA) and proposed an efficient optimizer for the unknown parameters in solar cells. Maouhoub et al. [41] used the least squares method to estimate the shunt resistance R_p, saturation current I_s, and photocurrent I_{ph} in a PV system, there was a small error between the experimental and measured results. Various conventional MPPT schemes have been proposed by many scholars, and the include the HC [4], incremental HC (INC) [80] and disturbance and observation methods [23]. These three types of methods have simple structures and low equipment requirements, but they cannot handle cases with partially shaded conditions (PSCs).

Weidong Xiao et al. [75] modified adaptive hill-climbing (MAHC) algorithm to avoid MPPT control tracking deviation. Hussam et al. [3] investigates the effect of measurement noise on the MPPT algorithm. Tey [70] presents the simulation and hardware implementation of incremental conductance algorithm using buck-boost converter and PIC18F4520 controller. Kenneth et al. [26] present status and perspectives on 100% renewable energy systems. Azadeh et al. [64] paper presents simulation and hardware implementation of incremental conductance (IncCond) MPPT used in solar array power systems with direct control method. FEMIA et al. [22] is prove that P&O may guarantee top-level efficiency, provided that a proper predictive and adaptive HC strategy is adopted. A modified P&O MPPT technique, applicable for PV systems proposed by Ahmed [1]. Piegari [60] propose adaptive perturb and observe algorithm for photovoltaic maximum power point tracking.

Because P—V curve often displays multiple peaks, including several local peaks (LP) and a global peak (GP), the traditional algorithms cannot determine whether a peak at a certain point is an LP or a GP. Miyatake et al. [42] proposed a scheme that uses a Fibonacci sequence to track the GP under PSCs. However, in this method, there is no guarantee that the GP point can be tracked due to the influence of the position of sun and cloud cover. Kobayashi et al. [37] proposed another two-stage method for GP tracking. In the first stage, the load line is used to detect the area of the GP, and in the second stage, the output will gradually converges according to the GP area scanned in the first stage.

The rest of this paper is arranged as follows, Section 2.1 introduces the detailed information for the circuit model including for the DDM and PV systems, additionally, the model parameter descriptions are given. Section 2.2 presents the fitness function proposed in this paper. Section 2.3 introduces the background theoretical knowledge for the PIO algorithm and the improvements to the TPIO algorithm. The superiority of the proposed algorithm to other algorithms is sown in Section 3. Section 3.2 describes how to use the improved algorithm to improve the accuracy of model parameters. Section 5 presents the final conclusions and the outlooks for future work based on the results of this paper.

2. Methods

2.1. Mathematical modeling of photovoltaic systems based on a double-diode model

The solar energy output of a system depends mainly on the sun, ambient temperature, and internal resistance of the system. However, the manufacturers of PV modules cannot provide sufficient ranges for all parameters, such as the open-circuit voltage (V_{oc}), short-circuit current (I_{sc}), maximum power current (I_{mp}), maximum power voltage (V_{mp}), and the ambient and short-circuit current coefficients α in a given PV module and the temperature coefficient β at which the module operates. The parameters provided by the manufacturer are set based on data obtained under standard test conditions (STCs), including the solar radiation intensity at 1000 W/m^2 and an initial temperature of 25 °C. The model developed by King [35,36] accurately predicts the energy output using a simple algebraic model, but this model requires parameter information that is not provided by the manufacturer. Later, data for many different PV module parameters were provided by the National Laboratory of Japan [38].

2.1.1. Double-diode model

An ideal photovoltaic cell uses light as a raw material to generate the output current (I_{ph}), but for electrical or optical reasons, I_{ph} is different from the actual current, which will result in current output loss [11,28]. Series and parallel effects are considered in the simple PV model as shown in Fig. 1; this model was proposed by Nesson [51], Duffe and Beckman [17]. The V—I characteristic of the current output I of the ideal model is calculated by utilizing the Shockley diode law and can be expressed by Eqs. (1)—(2).

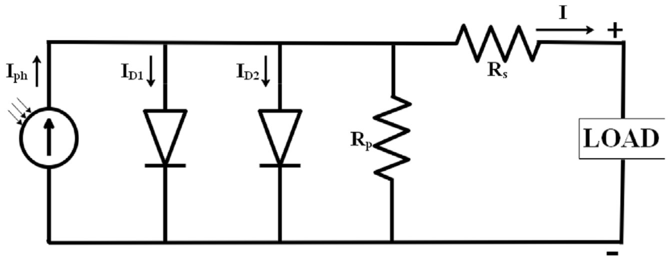


Fig. 1. Equivalent circuit of the double-diode model.

$$I = I_{ph} - I_d - I_R. \tag{1}$$

To consider the load current in space, the PV module current I is shunted with the other two diodes considering the shunt leakage resistance, and the relatively stable region of the emitter in the semiconductor and the high concentration region near the P—N junction can be considered in the DDM. The leakage effect current I_d of the DDM is expressed as shown in Eq. (2).

$$I_d = I_{d1} + I_{d2}, \tag{2}$$

$$\begin{cases} I_{d1} = I_{01}\left\{e^{\frac{q(V+R_sI)}{a_1KT}} - 1\right\} \\ I_{d2} = I_{02}\left\{e^{\frac{q(V+R_sI)}{a_2KT}} - 1\right\}, \end{cases} \tag{3}$$

where the charge of an electron q is 1.6 × $10^{-19}C$. T is the temperature of the model. The Boltzmann constant K is 1.3805 × $10^{-23}J/K$. I_{01} and I_{02} are leakage current. a_1 and a_2 are diode ideality factors. R_s is the series resistance.

A DDM resistor is used in series with an ohmic contact to avoid metal resistance (ohmic losses); the contact impurity concentration and the depth of the P—N junction affect current flow. A leakage current at the P—N junction indicates that shunt resistance exists in the DDM as given by Eq. (4).

$$I_R = \frac{V + IR_s}{R_{sh}}, \tag{4}$$

where V is the input voltage and R_{sh} is shunt resistance.

The above equations indicate that the series resistance R_s and shunt resistance R_{sh} impact the V—I characteristic curve of the DDM. The output voltage will also be affected by the series resistance R_{sh}, notably R_{sh} will reduce the output current I [6,11,50,65]. Eq. (1) is used to modify the equation for PV modules. The DDM terminal current can be determinged based on seven parameters, as given by Eq. (5).

$$I = I_{ph} - I_{01}\left(e^{\frac{q(V+RsI)}{a_1KT}} - 1\right) - I_{02}\left(e^{\frac{q(V+RsI)}{a_2KT}} - 1\right) - \frac{V + IR_s}{R_{sh}}, \tag{5}$$

In the parameter extraction model, the parameter set is: $\Phi = \left[I_{ph}, I_{01}, I_{02}, R_s, R_{sh}, a_1, a_2\right]$, and these parameters can be extracted to establish the V—I relation. In this section, $f(V_e, I_e, \Phi)$ is defined as:

$$f(V_e, I_e, \Phi) = \quad I_{ph} - I_{01}\left(e^{\frac{q(V_e+R_sI_e)}{a_1KT}} - 1\right) \\ -I_{02}\left(e^{\frac{q(V_e+R_sI_e)}{a_2KT}} - 1\right) - \frac{V_e + I_eR_s}{R_{sh}} - I_e. \qquad (6)$$

2.1.2. PV systems

The considered circuit diagram model includes series and parallel PV modules. The PV array is composed of multiple DDM modules and a PV module, as shown in the Fig. 2 [17]. The DDM parameters are transformed to represent the PV module. The PV module parameters are transformed based on topology with series and parallel cells, as shown in Table 1, where V_t can be expressed by $(aKT)/q$.

The seven parameter sets Φ have an impact on the PV module. In the PV array, N_s cells are coupled in series, and used to increase the output voltage; additionally, N_p cells are connected in parallel to increase the output current.

The PV array is composed of multiple solar cells shielded in series and parallel to prevent environmental changes from affecting the module. In the case of PSCs, the current source I_{ph} is described by Eq. (7).

$$I_{ph} = (I_{pv,STC} + K_I\Delta_T)\frac{G}{G_n}, \qquad (7)$$

where $I_{pv,STC}$ is the light-generated current under STC, K_I is the short-circuit current, which is provided by the manufacturer when a module leaves the factory; $\Delta_T = T - T_{STC}$, T is the actual temperature of the current cell; and T_{STC} is the nominal temperature of the current cell [in Kelvin]. G is the irradiation on the cell surface, G_n is the nominal irradiation and these parameters are given in watts per square meters. The light-generated current is approximately proportional to the intensity of sunlight.

Therefore, the equation for the output current of the PV array can be written as shown in Eq. (8):

$$I = \quad N_pI_{ph} - N_pI_{01}\left[\exp\left(\frac{1}{V_{t1}}\left(\frac{V}{N_s} + \frac{IR_s}{N_p}\right)\right) - 1\right] \\ -N_pI_{02}\left[\exp\left(\frac{1}{V_{t2}}\left(\frac{V}{N_s} + \frac{IR_s}{N_p}\right)\right) - 1\right] \\ -\frac{N_p}{R_p}\left(\frac{V}{N_s} + \frac{IR_s}{N_p}\right). \qquad (8)$$

Most PV arrays also have bypass diodes and reverse blocking diodes. A typical PV power generation system is composed in such a way that it can meet the power demand of load devices. In this paper, the effects of these two diodes on global power considered. It can be assumed that there are two arrays; one array is fully illuminated by light and the other is partially covered by shadows. In this case, the current flowing through the two arrays is the same because the arrays are connected in series, but the light absorbed by the second array produces less current than that absorbed by the first array. Fig. 3 shows the V—I characteristics of two single modules and the entire PV system. Notably, two MPPs, a GP and an LP, are shown. If there are multiple arrays connected, the uneven distribution of sunlight will lead to heterogeneity throughout the system, and LPs may be obtained instead of a global MPP. In this case, conventional methods of MPPT are difficult and time consuming to implement. Even if a method does converge, there is no guarantee that the identified point is a global MPP point. In this case, failure to identify the global MPP will cause the entire system to fail to reach an optimal state.

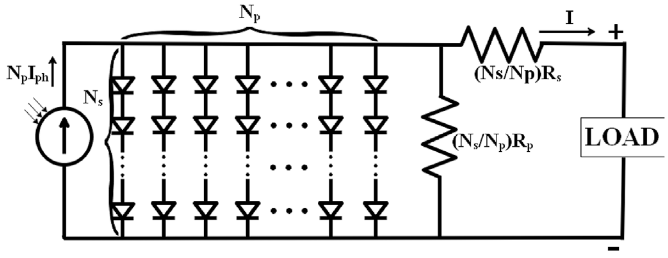


Fig. 2. Equivalent circuit of the PV array model.

Fig. 4 shows the PV curve of a commercial module with a 5 × 1 PV array configuration. The parameters of this particular array under STCs are shown in Table 2.

2.2. Fitness function

This paper uses the real measurement data set to extract the unknown parameter value inside the solar model, which is then used to track the MPPT of the PV systems. The information extracted from the real measurement data can more realistically calculate the maximum power of systems.

The purpose of model construction and parameter extraction is to reduce calculation errors and actual current levels under experimental conditions, thus, the seven parameters can be estimated based on the root mean square error (*RMSE*) [25] as the minimized objective function to optimize the Eq. (9).

$$RMSE = \sqrt{1 \Big/ N\sum_{i=1}^{N} f(V_e, I_e, \Phi)^2}, \qquad (9)$$

where N is the length of the V—I data curve, V_e and I_e are actual values, and Φ is the parameter that needs to be extracted.

The MPPT information for the PV systems was introduced in the previous section, in which the PV characteristics include multiple LPs. When two or more modules are connected in series, the corresponding PV array may be partially covered, and the terminal voltages of the covered and uncovered parts of the array will differ. Suppose there are two PV arrays: one that is covered and one that is completely illuminated by sunlight; their voltages are V_1 and V_2, and the power output of the PV system is P. Extending this problem to multiple PV arrays will transform the MPPT problem into a multidimensional optimization problem, and the voltage of each array must be controlled. Generally, if PV systems contain multiple arrays, the terminal voltage (V_1, V_2, …, V_N) for each individual array must be controlled. Therefore, the terminal voltage of each PV array is combined to form an N-dimensional vector (10).

$$v^t = [V_1^t, V_2^t, \ldots, V_i^t, \ldots, V_N^t], \qquad (10)$$

where N is the number of PV arrays in the system, which can be used to determine the size of the parameter vector. Here, the fitness function P is the sum of the power generated by each power array and can be expressed as shown in Eq. (11).

$$P = \sum_{i=1}^{N} V_i \times I_i, \qquad (11)$$

where V_i and I_i are the terminal voltage and output current of each array, respectively.

Based on the fundamental fitness function for PV cells given in Eq. (9) and the fitness function used in the MPPT method given in Eq. (11), we propose a new fitness function that can be used for

Table 1
PV module parameter conversion based on series and parallel cells.

No.	Parameters of PV cell	Parameters of PV array (N_S cells connected in series)	Parameters of PV array (N_p cells connected in parallel)
1.	I_{ph}	I_{ph}	N_pI_{ph}
2.	V_t	N_sV_t	V_t
3.	R_s	N_sR_s	R_s/N_p
4.	R_{sh}	N_sR_{sh}	R_{sh}/N_p

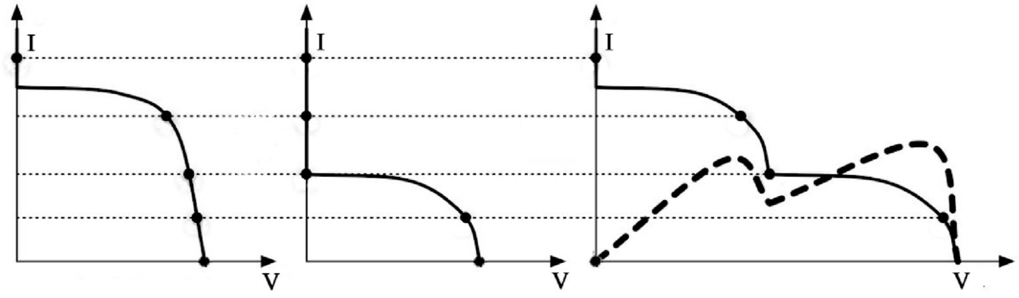


Fig. 3. Characteristics of PV systems connected in series including fully illuminated, partially covered.

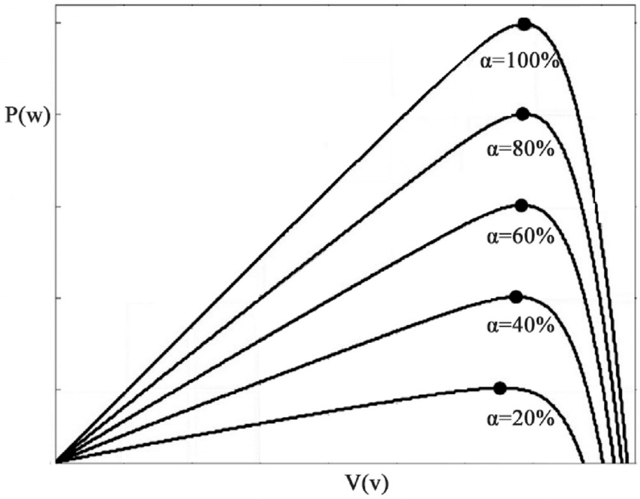


Fig. 4. P–V curves for different irradiation levels.

Table 2
Parameters of PV system at STC: temperature = 25°C, insolation = 1000W/m^2.

Parameters Name	value
Peak Power (W), P_{pp}	60
Peak power voltage (V), V_{ppv}	17.1
Peak power current (A), I_{ppc}	3.5
Open circuit voltage (V), V_{ocv}	21.1
Short circuit current (A), I_{sc}	3.8
Temperature coefficient of current $(mA/^{\circ}C), K_I$	0.003
Temperature coefficient of voltage $(mV/^{\circ}C), K_V$	−0.08
Number of series cells, N_s	36

optimization as shown in Eq. (12).

$$fitness = |RMSE| + \left|\frac{1}{P}\right|. \tag{12}$$

2.3. Pigeon-inspired optimization

The PIO algorithm is a new bio-inspired swarm intelligence algorithm proposed by Duan based on homing behavior of pigeons. The pigeon is one of the most studied birds, especially the study of homing behavior. Research on the behavior of pigeons capable of different performance skills in different magnetic fields has shown that the behavior of pigeons that can find home in various situations is dependent on the nose in their body structure, without which the nose with its unique signal of iron ore particles would not have the ability to discern direction. There is evidence that pigeons have a special system where signals from magnetite particles are transmitted from their noses to the cerebral cortex via their trigeminal nerves. Studies of pigeon behavior have shown that pigeons can follow some landmarks, such as tall buildings, rivers, etc., instead of going directly to their destination. The ability of pigeons to distinguish the difference in solar height between the release point and the base also explains the role of the sun in guiding the pigeons to navigate home. The PIO algorithm is inspired by the above behavioral characteristics of pigeons. the advantages of the PIO algorithm, such as high convergence and fast computing speed, have solved a large number of engineering control problems, and it also performs well in this paper for wind turbine parameter extraction.

However, because of its advantage of fast convergence, PIO can also easily cause the disadvantage of falling into local optimum. To overcome this drawback of PIO, an improved PIO algorithm based on Taguchi method (TPIO) is proposed for the first time in this paper. Compared with the PIO algorithm, the TPIO algorithm has better global search capability and can also effectively avoid the problem of the algorithm falling into local optimum. The TPIO algorithm proposed in this paper is compared with several other algorithms for experiments. The results of the comparison experiments prove the feasibility and effectiveness of the method.

Pigeons generally use three homing tools to help them find their way home: the magnetic field, the sun, and landmarks. Pigeons can form a map in their minds after sensing the magnetic field through a magnetic receiver. Then, the pigeon adjusts its direction of movement based on the sun as a disk. As the pigeons get closer to their destination, they become less dependent on the magnetic field and the sun, and instead use landmarks to guide their movement. If the pigeons are more familiar with the destination, then they will fly directly to it, and if they are unfamiliar with it, then they will follow the familiar pigeons.

2.3.1. Initialization stage

In the PIO algorithm, virtual pigeons are introduced into the mathematical model. In the map and compass operator, the rule followed is to use *Pos* and *Vel* to represent the position and velocity of the pigeon, and after each iteration, *Pos* and *Vel* in *D* dimensions can be calculated using the following equation.

$$\begin{aligned} Pos &= [Pos_1, Pos_2, \ldots, Pos_N] \\ \text{Vel} &= [Vel_1, Vel_2, \ldots, Vel_N] \end{aligned} \tag{13}$$

Like most heuristic algorithms, the PIO algorithm is guided by an initialized population generated independently of the same distribution all over a uniform random distribution. Each individual is represented by a vector. The initialization process of the entire population is executed as following:

$$Pos = lb + \text{ rand } \cdot (ub - lb) \tag{14}$$

where, *lb* and *ub* are the search boundary of decision variables *Pos*, and rand is a random number defined in the range [0, 1].

2.3.2. Map and compass operator

The PIO algorithm begins with a random set of initial solutions and updates the position and velocity of each pigeon according to a specified search rule. The position Pos_i and velocity Vel_i of the pigeon i at the t-th iteration can be expressed as follows:

$$\begin{gathered} Vel_i^t = Vel_i^{t-1} \cdot e^{-R \cdot t} + \text{ rand } \cdot \left(Pos_g^t - Pos_i^{t-1}\right) \\ Pos_i^t = Pos_i^{t-1} + Pos_i^t \end{gathered} \tag{15}$$

where, R is the factor of map and compass, and Pos_g^t is the best position, which can be got by comparing all position of pigeons in current iterations.

Ensure the best location for each pigeon by using the map and compass factor. Comparing the position information of all pigeons, the movement process of each pigeon is influenced by the global optimum Pos_g and its own position of the previous generation. The function value of the observation function $e^{-R \cdot t}$ is decreasing as the iterations continue, in other words, the pigeon receives a decreasing influence of the speed of the previous generation during its movement.

2.3.3. Landmark operator

In the landmark operator, the pigeon will gradually approach its destination, and if the map and compass continue to be used to guide the pigeon, its flight is no longer extremely useful. The pigeon will use the landmark operator to guide its flight. At this stage, the number of pigeons N is reduced by half for each iteration. The pigeons that know the way will fly directly to their destination, while the pigeons that don't know the way will follow the pigeons that know the way and continue their flight. Let Pos_c^t be the centroid of the position after the t-th iteration, provided that each pigeon is considered to fly directly to its destination. The rule for updating the position of pigeon i at the t-th iteration is given by the following equation.

$$\begin{gathered} N^t = \frac{N^{t-1}}{2} \\ Pos_c^t = \frac{\sum Po\underset{i}{\overset{t}{s}} \cdot \text{ Fitness } \left(Pos_i^{t-1}\right)}{N^t \sum Fitness\left(Po\,\underset{i}{\overset{t-1}{s}}\right)} \\ Pos_i^t = Pos_i^{t-1} + rand \cdot \left(Pos_c^t - Pos_i^{t-1}\right) \end{gathered} \tag{16}$$

where Pos_c^t is the center of all pigeon's position at the iteration t, $Fitness\left(Pos_i^{t-1}\right)$ is the quality of each individual, it can be calculated by fitness function. If it is a maximum value problem, let Fitness $\left(Pos_i^t\right) = fitnessfunction\left(Pos_i^t\right)$. If it is a minimum value problem, then let Fitness $(Pos_i(t)) = 1/\left(fitnessfunction\left(Pos_i^t\right) + \epsilon\right)$. For the minimization problem, an ϵ is added to prevent the denominator from becoming nonsensical. The Flow chart of PIO is shown in Fig. 5.

2.4. Taguchi method

Experimental design is a mathematical theory and method for developing experimental protocols economically and scientifically for effective statistical analysis of experimental data, based on the theory of probability theory and mathematical statistics. The basic idea of Taguchi method was made famous by Dr. G. Taguchi of Japan.

The Taguchi method is widely used for production line design and process conditions because it allows for high quality product output with few computational resources. One of the important tools in Taguchi method is called orthogonal array, which can be adopted by PIO algorithm to improve its own performance. In the TPIO algorithm, just taken two-level orthogonal array to join in the optimize process. In the following article, we will first describe the concept of the two-level orthogonal array of Taguchi method. First define an orthogonal array, because each column of this array will represent the value of a considered factor, and in addition the factors in this orthogonal array can be operated independently. For each row of the orthogonal array, the experiment is performed once, and each column represents the number of parameters to be optimized and the values to be taken. The two-level array used in this paper can be described as follows.

$$L_n\left(2^{n-1}\right) \tag{17}$$

where $n = 2^k$ and k is a positive integer greater than 1, while $n - 1$ represents the number of columns in a two-level orthogonal array. For example, suppose we design an orthogonal array table containing eleven dimensions, with the aim of finding the best combination of the dimensional combinations. In that case, the orthogonal array table of $L_{12}(2^{11})$ is constructed as shown in Table 3.

The elements in the orthogonal matrix described in Table 3 and it indicate which level value should be taken for each dimension under each test. We can see that the data in Table 3 are all values containing only "0″ or "1″, so in the example it is a two-level example. When the value in the table is "0″, the value represented by one level is taken, and when the value in the table is "1", the value represented by the other level is taken. The orthogonal matrix approach was proposed by Taguchi to effectively reduce the number of experimental runs. The steps in Table 3 require 2^{11} number of experiments to explore all combinations of experiments, but in Taguchi method reduces the number of experiments to 12 to complete all experiments.

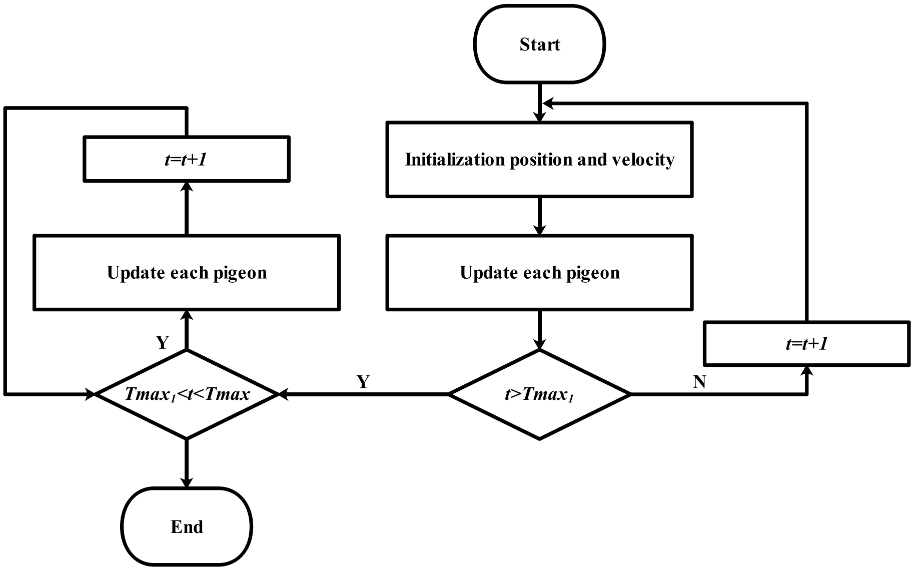


Fig. 5. The Flow chart of PIO.

Table 3
This is a two-level orthogonal array.

Experiment number	Dimensional Factors										
	A	B	C	D	E	F	G	H	I	J	K
1	0	0	0	1	0	0	1	0	1	1	1
2	0	0	1	0	0	1	0	1	1	1	0
3	0	0	1	0	1	1	1	0	0	0	1
4	0	1	0	0	1	0	1	1	1	0	0
5	0	1	0	1	1	1	0	0	0	1	0
6	0	1	1	1	0	0	0	1	0	0	1
7	1	0	0	0	1	0	0	1	0	1	1
8	1	0	0	1	0	1	1	1	0	0	0
9	1	0	1	1	1	0	0	0	1	0	0
10	1	1	0	0	0	1	0	0	1	0	1
11	1	1	1	0	0	0	1	0	0	1	0
12	1	1	1	1	1	1	1	1	1	1	1

2.5. *Pigeon-inspired optimization based on Taguchi method (TPIO)*

As can be seen from sections 2.3.2 and 2.3.3, the PIO algorithm contains two parts, a map and compass operator, and a landmark operator. The former is the exploration phase and the latter is the exploitation phase. The advantage of this is that the entire algorithm can be operated in two steps, avoiding the capability that similar PSO needs to balance exploration and exploitation. However, there are still some problems with PIO algorithm, the timing of performing exploration and exploitation iterations, the size of the exploration or exploitation capacity, and the ability to jump out of the local optimum. In this paper, in order to overcome the shortcomings of the PIO algorithm, we introduce the orthogonal matrix of Taguchi method into the landmark operator, and then the PIO algorithm can get better converge to a more desirable value and enhance the exploitation capability of the algorithm.

Taguchi method can be very effective in reducing the number of runs of the experiment. The algorithm is proved to be in the development stage when the landmark operator begins to execute. However, according to the mathematical model of the PIO algorithm, it can be analyzed that the number of pigeons decreases as the iterations proceed, which may lead to the risk of the algorithm falling into a local optimum. With the introduction of the orthogonal matrix in Taguchi method, it is possible to facilitate the convergence of the algorithm to a more superior value based on the global optimum and the local central value. The operation to enhance the convergence of the algorithm as follows:

Step 1: Two sets of candidate set velocities are generated, denoted by $tv_{1,j}$ and $tv_{2,j}$, and the specific representation is shown in Eq. (18).

$$\begin{aligned} tv_{1,j}^{t} &= Vel_{i,j}^{t-1} + r_1 \cdot c_1 \cdot \left[Pos_{g,j}^{t-1} - Pos_{i,j}^{t-1}\right], \\ tv_{2,j}^{t} &= Vel_{i,j}^{t-1} + r_1 \cdot c_1 \cdot \left[Pos_{c,j}^{t-1} - Pos_{i,j}^{t-1}\right], j = 1, 2, \ldots, D \end{aligned} \tag{18}$$

where D represents the dimensionality of the problem, Pos_g represents the global optimal solution found by the algorithm so far, Pos_c represents the position of the pigeon at the center of the landmark operator stage, i represents the i-th pigeon, $r1$ is a random value in the range [0–1], and $c1$ is an inertia constant used to control the size of the resulting velocity set. A series of velocity sets are created using candidate velocities and Taguchi orthogonal matrices as shown in Eq. (19).

$$Vel_{\text{set }, j}^{t} = \begin{cases} tv_{1,j}^{t}, & \text{if the element is " 0 "} \\ tv_{2,j}^{t}, & \text{otherwise} \end{cases} \tag{19}$$

Step 2: Take a velocity set $Vel_{\text{set }, j}^{t}$ to update the original velocity $Vel_{i,j}^{t}$. If the updated velocity exceeds the maximum velocity limit, make it equal to the maximum value.

$$Vel_{i,j}^{t} = \begin{cases} Vel_{\max}, \ \text{if} \left[Vel_{i,j}^{t-1} + Vel_{\text{set }, j}^{t}\right] \\ \text{exceed the maximum velocity boundary} \\ Vel_{i,j}^{t-1} + Vel_{\text{set }, j}^{t}, \ \text{otherwise} \end{cases} \tag{20}$$

and then update the position $Pos_{i,j}^{t}$ of pigeon, as shown follows Eq. (21):

$$Pos_{i,j}^{t} = Pos_{i,j}^{t-1} + Vel_{i,j}^{t} \tag{21}$$

This pigeon position is used as a candidate solution and then calculated using the fitness function. The cumulative fitness value is the most suitable combination of the orthogonal matrix to form the speed factor.

Step 3: Move the position of the pigeon using Eq. (21) to update its candidate set.

The above describes the operation mechanism of the TPIO algorithm, and let us describe this selection process with an example. For heuristic optimization algorithms, we should first assume a fitness function, where we assume the use of a fitness function (22) to evaluate the merits of candidate solutions.

$$fitness(Pos) = \sum_{j=1}^{D} \frac{1}{Pos_j} \tag{22}$$

Based on the above fitness function, it is the purpose of our work to minimize the value of this function. Before performing the calculation, we first initialize the position and velocity of the pigeon, here for the convenience of the calculation, we set both the position of the pigeon at 0 and the velocity at 1. According to the design of the orthogonal matrix table of $L_{12}(2^{11})$, we set the dimension of optimization to 11. The specific properties of the initialized pigeon are shown in Table 4.

After initializing the position and velocity of the pigeon, we should generate the candidate velocity solution sets $tv_{1,j}$ and $tv_{2,j}$ according to Eq. (20). Here we assume that the generated candidate velocity solution sets are shown in Table 5.

According to the orthogonal matrix shown in Table 3, the number of experiments needed to be performed in this paper is 12. According to Table 3, the set of 12 solutions is formed by using the values of tv_1 and tv_2 represented by “0″ and “1”. The experimental results and the final calculation results are shown in Table 6.

According to Table 6, it is known that when the element of the orthogonal matrix in Table 3 takes “0″ in a dimension, it represents the level value of “0”. Table 4 shows that when the level is “0″, the value of tv_1 is taken in the corresponding dimension. Then, we update the speed and position information in the next generation position matrix reference Table 5 according to Eq. (10). Then, the fitness value of each experiment in the orthogonal matrix is calculated according to the fitness function 11, and then the cumulative sum of each dimension is calculated according to the choice of the orthogonal matrix as “0″ or “1”. The final selection of candidate velocities is given in Table 7.

Table 4
The initial position and speed of the pigeon.

	A	B	C	D	E	F	G	H	I	J	K
Pos	0	0	0	0	0	0	0	0	0	0	0
$Vel_{i,j}^{t-1}$	1	1	1	1	1	1	1	1	1	1	1

Table 5
The candidate velocity sets.

	A	B	C	D	E	F	G	H	I	J	K
tv_1	5	7	0	6	4	1	3	0	0	4	3
tv_2	1	0	4	0	3	6	0	1	6	5	1

3. Results

In this section, the main purpose is to evaluate the performance of TPIO in the maximum power tracking of the test function and solar parameter extraction. First, compare TPIO with other six algorithms in the CEC2014 test function set. Then the TPIO algorithm is applied to parameter extraction and power tracking. The experiment runs on a personal desktop computer with Windows 10 operating system, 24 GB memory, Intel(R) Core(TM) i5-8500 CPU @ 3.00 GHz 3.00 GHz, and the software environment is Matlab2020b.

3.1. Benchmark functions and result analysis

In this section, we use the CEC2014 test function set for global optimization to verify the performance of the TPIO algorithm. The CEC2014 function set contains 30 benchmark functions, including

Table 7
The candidate velocity sets.

Candidate velocity	Considered factors										
	A	B	C	D	E	F	G	H	I	J	K
vk,d(t)	5	7	4	6	4	6	3	1	6	5	3

Table 8
Details of the parameters of the algorithm.

Algorithm	Parameters settings
PSO [20]	$c_1 = 2$, $c_2 = 2$, $w = 0.9$ to 0.4
DA [46]	$w = 0.9$ to 0.4, $my_c = 0.1$ to -0.1
GWO [21]	$a = 2$ to 0
NPIO	$threshold = 10$
PIO [16]	$R = 0.2$
SCA [9]	$a = 2$

Table 6
The experiment calculated fitness values.

Experiment number	Updated coordinates of the artificial agent											Fitness value
	A	B	C	D	E	F	G	H	I	J	K	
1	0 + 1+5	0 + 1+7	0 + 1+0	0 + 1+0	0 + 1+4	0 + 1+1	0 + 1+0	0 + 1+0	0 + 1+6	0 + 1+5	0 + 1+1	5.8012
2	0 + 1+5	0 + 1+7	0 + 1+4	0 + 1+6	0 + 1+4	0 + 1+6	0 + 1+3	0 + 1+1	0 + 1+6	0 + 1+5	0 + 1+3	2.2869
3	0 + 1+5	0 + 1+7	0 + 1+4	0 + 1+6	0 + 1+3	0 + 1+6	0 + 1+0	0 + 1+0	0 + 1+0	0 + 1+4	0 + 1+1	4.7274
4	0 + 1+5	0 + 1+0	0 + 1+0	0 + 1+6	0 + 1+3	0 + 1+1	0 + 1+0	0 + 1+1	0 + 1+6	0 + 1+4	0 + 1+3	5.1524
5	0 + 1+5	0 + 1+0	0 + 1+0	0 + 1+0	0 + 1+3	0 + 1+6	0 + 1+3	0 + 1+0	0 + 1+0	0 + 1+5	0 + 1+3	6.2262
6	0 + 1+5	0 + 1+0	0 + 1+4	0 + 1+0	0 + 1+4	0 + 1+1	0 + 1+3	0 + 1+1	0 + 1+0	0 + 1+4	0 + 1+1	5.5167
7	0 + 1+1	0 + 1+7	0 + 1+0	0 + 1+6	0 + 1+3	0 + 1+1	0 + 1+3	0 + 1+1	0 + 1+0	0 + 1+5	0 + 1+1	4.9345
8	0 + 1+1	0 + 1+7	0 + 1+0	0 + 1+0	0 + 1+4	0 + 1+6	0 + 1+0	0 + 1+1	0 + 1+0	0 + 1+4	0 + 1+3	5.9179
9	0 + 1+1	0 + 1+7	0 + 1+4	0 + 1+0	0 + 1+3	0 + 1+1	0 + 1+3	0 + 1+0	0 + 1+6	0 + 1+4	0 + 1+3	4.4179
10	0 + 1+1	0 + 1+0	0 + 1+0	0 + 1+6	0 + 1+4	0 + 1+6	0 + 1+3	0 + 1+0	0 + 1+6	0 + 1+4	0 + 1+1	5.0786
11	0 + 1+1	0 + 1+0	0 + 1+4	0 + 1+6	0 + 1+4	0 + 1+1	0 + 1+0	0 + 1+0	0 + 1+0	0 + 1+5	0 + 1+3	5.9595
12	0 + 1+1	0 + 1+0	0 + 1+4	0 + 1+0	0 + 1+3	0 + 1+6	0 + 1+0	0 + 1+1	0 + 1+6	0 + 1+5	0 + 1+1	5.4024
Accumulated of tv_1	29.7107	28.0857	33.1107	28.1393	30.5607	31.7821	28.4607	32.2107	33.2821	30.8107	29.9607	
Accumulated of tv_1	31.7107	33.3357	28.3107	33.2821	30.8607	29.6393	32.9607	29.2107	28.1393	30.6107	31.4607	
The chosen candidate	tv_1	tv_1	tv_2	tv_1	tv_1	tv_2	tv_1	tv_2	tv_2	tv_2	tv_1	

Table 9
Summary of the CEC2014 test functions.

Algorithm Description	No	Function express	Minimum value
Unimodal Functions	*f*1	Rotated High Conditioned Elliptic Function	100
	*f*2	Rotated Bent Cigar Function	200
	*f*3	Rotated Discus Function	300
Simple Multimodal Functions	*f*4	Shifted and Rotated Rosenbrock's Function	400
	*f*5	Shifted and Rotated Ackley's Function	500
	*f*6	Shifted and Rotated Weierstrass Function	600
	*f*7	Shifted and Rotated Griewank's Function	700
	*f*8	Shifted Rastrigin's Function	800
	*f*9	Shifted and Rotated Rastrigin's Function	900
	*f*10	Shifted Schwefel's Function	1000
	*f*11	Shifted and Rotated Schwefel's Function	1100
	*f*12	Shifted and Rotated Katsuura Function	1200
	*f*13	Shifted and Rotated HappyCat Function	1300
	*f*14	Shifted and Rotated HGBat Function	1400
	*f*15	Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	1500
	*f*16	Shifted and Rotated Expanded Scaffer's F6 Function	1600
Hybrid Function 1	*f*17	Hybrid Function 1 (N = 3)	1700
	*f*18	Hybrid Function 2 (N = 3)	1800
	*f*19	Hybrid Function 3 (N = 4)	1900
	*f*20	Hybrid Function 4 (N = 4)	2000
	*f*21	Hybrid Function 5 (N = 5)	2100
	*f*22	Hybrid Function 6 (N = 5)	2200
Composition Functions	*f*23	Composition Function 1 (N = 5)	2300
	*f*24	Composition Function 2 (N = 3)	2400
	*f*25	Composition Function 3 (N = 3)	2500
	*f*26	Composition Function 4 (N = 5)	2600
	*f*27	Composition Function 5 (N = 5)	2700
	*f*28	Composition Function 6 (N = 5)	2800
	*f*29	Composition Function 7 (N = 3)	2900
	*f*30	Composition Function 8 (N = 3)	3000
Search Range: $[-100,100]^D$			

Table 10
TPIO and PSO, DA, GWO comparison performance.

D = 30	PSO			DA			GWO		
	Mean	Std	Min	Mean	Std	Min	Mean	Std	Min
*f*1	6.80×10^{07}	7.30×10^{07}	1.43×10^{07}	3.80×10^{08}	2.17×10^{08}	1.05×10^{08}	3.86×10^{07}	2.48×10^{07}	7.62×10^{06}
*f*2	9.99×10^{09}	4.30×10^{09}	1.80×10^{09}	1.58×10^{10}	8.60×10^{09}	4.14×10^{09}	1.12×10^{09}	9.78×10^{08}	1.68×10^{08}
*f*3	1.67×10^{04}	1.80×10^{04}	1.48×10^{03}	7.42×10^{04}	2.51×10^{04}	2.72×10^{04}	2.59×10^{04}	8.34×10^{03}	1.18×10^{04}
*f*4	1.05×10^{03}	3.69×10^{02}	5.52×10^{02}	3.58×10^{03}	1.80×10^{03}	1.02×10^{03}	5.64×10^{02}	4.86×10^{01}	4.85×10^{02}
*f*5	5.21×10^{02}	1.53×10^{01}	5.20×10^{02}	5.21×10^{02}	1.81×10^{01}	5.20×10^{02}	5.21×10^{02}	4.41×10^{-02}	5.21×10^{02}
*f*6	6.19×10^{02}	3.46×10^{00}	6.14×10^{02}	6.35×10^{02}	3.63×10^{00}	6.26×10^{02}	6.12×10^{02}	2.92×10^{00}	6.08×10^{02}
*f*7	7.89×10^{02}	3.51×10^{01}	7.43×10^{02}	9.10×10^{02}	8.08×10^{01}	7.17×10^{02}	7.13×10^{02}	1.28×10^{01}	7.01×10^{02}
*f*8	8.69×10^{02}	1.97×10^{01}	8.34×10^{02}	1.03×10^{03}	4.95×10^{01}	9.54×10^{02}	8.74×10^{02}	2.08×10^{01}	8.42×10^{02}
*f*9	1.02×10^{03}	1.87×10^{01}	9.85×10^{02}	1.15×10^{03}	6.17×10^{01}	9.99×10^{02}	9.92×10^{02}	2.33×10^{01}	9.59×10^{02}
*f*10	3.94×10^{03}	6.50×10^{02}	2.30×10^{03}	6.19×10^{03}	1.18×10^{03}	3.89×10^{03}	3.24×10^{03}	5.74×10^{02}	2.44×10^{03}
*f*11	4.41×10^{03}	6.98×10^{02}	3.22×10^{03}	7.35×10^{03}	9.53×10^{02}	4.57×10^{03}	3.57×10^{03}	5.99×10^{02}	2.56×10^{03}
*f*12	1.20×10^{03}	6.02×10^{-01}	1.20×10^{03}	1.20×10^{03}	7.08×10^{-01}	1.20×10^{03}	1.20×10^{03}	1.08×10^{00}	1.20×10^{03}
*f*13	1.30×10^{03}	9.36×10^{-01}	1.30×10^{03}	1.30×10^{03}	7.36×10^{-01}	1.30×10^{03}	1.30×10^{03}	2.91×10^{-01}	1.30×10^{03}
*f*14	1.43×10^{03}	2.03×10^{01}	1.40×10^{03}	1.49×10^{03}	3.10×10^{01}	1.44×10^{03}	1.40×10^{03}	3.70×10^{00}	1.40×10^{03}
*f*15	1.11×10^{04}	3.10×10^{04}	1.51×10^{03}	1.19×10^{04}	2.41×10^{04}	1.52×10^{03}	1.53×10^{03}	2.25×10^{01}	1.50×10^{03}
*f*16	1.61×10^{03}	3.57×10^{-01}	1.61×10^{03}	1.61×10^{03}	4.20×10^{-01}	1.61×10^{03}	1.61×10^{03}	6.40×10^{-01}	1.61×10^{03}
*f*17	2.15×10^{06}	1.38×10^{06}	1.20×10^{05}	1.44×10^{07}	1.12×10^{07}	1.15×10^{06}	1.04×10^{06}	1.28×10^{06}	1.28×10^{05}
*f*18	2.19×10^{07}	8.45×10^{07}	2.42×10^{03}	2.84×10^{08}	3.27×10^{08}	2.22×10^{03}	5.70×10^{06}	1.76×10^{07}	2.28×10^{03}
*f*19	1.93×10^{03}	3.16×10^{01}	1.91×10^{03}	2.00×10^{03}	5.82×10^{01}	1.94×10^{03}	1.92×10^{03}	7.23×10^{00}	1.91×10^{03}
*f*20	9.04×10^{03}	1.08×10^{04}	2.91×10^{03}	4.60×10^{04}	2.93×10^{04}	1.25×10^{04}	1.72×10^{04}	6.90×10^{03}	2.70×10^{03}
*f*21	6.90×10^{05}	8.42×10^{05}	5.20×10^{04}	6.65×10^{06}	4.93×10^{06}	3.82×10^{05}	4.73×10^{05}	1.19×10^{06}	3.05×10^{04}
*f*22	2.68×10^{03}	2.80×10^{02}	2.28×10^{03}	$3.22 \times 10^{0}3$	2.90×10^{02}	2.60×10^{03}	2.50×10^{03}	1.15×10^{02}	2.35×10^{03}
*f*23	2.66×10^{03}	2.66×10^{01}	2.62×10^{03}	2.77×10^{03}	6.16×10^{01}	2.70×10^{03}	2.63×10^{03}	9.49×10^{00}	2.62×10^{03}
*f*24	2.65×10^{03}	1.13×10^{01}	2.63×10^{03}	2.66×10^{03}	1.21×10^{01}	2.64×10^{03}	2.60×10^{03}	1.01×10^{-03}	2.60×10^{03}
*f*25	2.71×10^{03}	4.52×10^{00}	2.71×10^{03}	2.74×10^{03}	1.30×10^{01}	2.72×10^{03}	2.70×10^{03}	3.19×10^{00}	2.70×10^{03}
*f*26	2.71×10^{03}	2.58×10^{01}	2.70×10^{03}	2.70×10^{03}	1.14×10^{00}	2.70×10^{03}	2.74×10^{03}	4.88×10^{01}	2.70×10^{03}
*f*27	3.49×10^{03}	3.07×10^{02}	3.11×10^{03}	3.37×10^{03}	9.32×10^{01}	3.19×10^{03}	3.33×10^{03}	1.36×10^{02}	3.11×10^{03}
*f*28	4.40×10^{03}	2.71×10^{02}	3.89×10^{03}	4.28×10^{03}	2.95×10^{02}	3.89×10^{03}	3.27×10^{03}	4.71×10^{01}	3.21×10^{03}
*f*29	7.27×10^{06}	7.34×10^{06}	7.56×10^{04}	1.36×10^{06}	6.96×10^{05}	3.10×10^{03}	3.11×10^{03}	5.07×10^{00}	3.10×10^{03}
*f*30	1.08×10^{05}	8.54×10^{04}	1.38×10^{04}	1.86×10^{05}	9.45×10^{04}	5.99×10^{04}	3.35×10^{03}	1.32×10^{02}	3.23×10^{03}
TPIO win	**24**			**28**			**18**		

Table 11
TPIO and NPIO, PIO, SCA comparison performance.

D = 30	NPIO			PIO			SCA			TPIO		
	Mean	Std	Min	Mean	Std	Min	Mean	Std	Min	Mean	Std	Min
*f*1	6.72×10^{07}	3.93×10^{07}	8.22×10^{06}	4.17×10^{08}	2.56×10^{08}	9.23×10^{07}	4.33×10^{08}	2.18×10^{08}	1.72×10^{08}	2.40×10^{07}	2.62×10^{07}	3.61×10^{06}
*f*2	1.49×10^{09}	1.25×10^{09}	9.76×10^{07}	1.23×10^{10}	6.61×10^{09}	2.43×10^{09}	2.54×10^{10}	4.37×10^{09}	1.67×10^{10}	5.06×10^{08}	9.38×10^{08}	1.75×10^{06}
*f*3	4.98×10^{04}	8.24×10^{03}	3.68×10^{04}	2.37×10^{05}	1.18×10^{05}	9.85×10^{04}	5.07×10^{04}	1.16×10^{04}	2.94×10^{04}	1.52×10^{04}	6.57×10^{03}	2.36×10^{03}
*f*4	6.63×10^{02}	6.34×10^{01}	5.52×10^{02}	2.89×10^{03}	2.01×10^{03}	8.25×10^{02}	2.23×10^{03}	6.91×10^{02}	1.17×10^{03}	5.83×10^{02}	7.91×10^{01}	4.89×10^{02}
*f*5	5.21×10^{2}	1.53×10^{-01}	5.20×10^{02}	5.21×10^{02}	1.35×10^{-01}	5.20×10^{02}	5.21×10^{02}	5.89×10^{-02}	5.21×10^{02}	5.21×10^{02}	2.01×10^{-01}	5.20×10^{02}
*f*6	6.27×10^{02}	3.11×10^{00}	6.20×10^{02}	6.31×10^{02}	4.18×10^{00}	6.21×10^{02}	6.38×10^{02}	1.55×10^{00}	6.34×10^{02}	6.21×10^{02}	2.77×10^{00}	6.15×10^{02}
*f*7	7.15×10^{02}	1.05×10^{01}	7.03×10^{02}	8.25×10^{02}	6.30×10^{01}	7.48×10^{02}	8.94×10^{02}	3.73×10^{01}	8.39×10^{02}	7.03×10^{02}	1.27×10^{00}	7.01×10^{02}
*f*8	9.58×10^{02}	3.08×10^{01}	8.81×10^{02}	1.01×10^{03}	4.37×10^{01}	9.07×10^{02}	1.08×10^{03}	2.04×10^{01}	1.04×10^{03}	8.71×10^{02}	1.80×10^{01}	8.37×10^{02}
*f*9	1.08×10^{03}	3.55×10^{01}	1.01×10^{03}	1.16×10^{03}	5.91×10^{01}	1.05×10^{03}	1.20×10^{03}	2.14×10^{01}	1.14×10^{03}	1.00×10^{03}	2.51×10^{01}	9.56×10^{02}
*f*10	5.25×10^{03}	6.33×10^{02}	3.82×10^{03}	6.71×10^{03}	1.15×10^{03}	3.65×10^{03}	7.83×10^{03}	5.06×10^{02}	6.76×10^{03}	2.86×10^{03}	5.02×10^{02}	1.82×10^{03}
*f*11	5.88×10^{03}	7.98×10^{02}	4.17×10^{03}	6.77×10^{03}	8.69×10^{02}	4.98×10^{03}	8.90×10^{03}	4.62×10^{02}	7.98×10^{03}	4.49×10^{03}	7.41×10^{02}	3.19×10^{03}
*f*12	1.20×10^{03}	4.28×10^{-01}	1.20×10^{03}	1.20×10^{03}	5.05×10^{-01}	1.20×10^{03}	1.20×10^{03}	2.58×10^{-01}	1.20×10^{03}	1.20×10^{03}	4.38×10^{-01}	1.20×10^{03}
*f*13	1.30×10^{03}	1.17×10^{-01}	1.30×10^{03}	1.30×10^{03}	1.32×10^{00}	1.30×10^{03}	1.30×10^{03}	4.20×10^{-01}	1.30×10^{03}	1.30×10^{03}	1.16×10^{-01}	1.30×10^{03}
*f*14	1.40×10^{03}	2.59×10^{00}	$1.40 \times 10^{\circ 3}$	1.46×10^{03}	2.98×10^{01}	1.41×10^{03}	1.47×10^{03}	1.22×10^{01}	1.45×10^{03}	1.40×10^{03}	3.71×10^{01}	1.40×10^{03}
*f*15	1.59×10^{03}	5.95×10^{01}	1.52×10^{03}	3.92×10^{04}	8.20×10^{04}	1.76×10^{03}	8.54×10^{03}	4.48×10^{03}	1.95×10^{03}	1.53×10^{03}	8.79×10^{00}	1.51×10^{03}
*f*16	1.61×10^{03}	5.94×10^{-01}	1.61×10^{03}	1.61×10^{03}	4.04×10^{-01}	1.61×10^{03}	1.61×10^{03}	2.48×10^{-01}	1.61×10^{03}	1.61×10^{03}	6.63×10^{-01}	1.61×10^{03}
*f*17	2.52×10^{06}	2.57×10^{06}	3.39×10^{05}	2.54×10^{07}	2.10×10^{07}	9.75×10^{05}	1.37×10^{07}	1.30×10^{07}	2.15×10^{06}	1.23×10^{06}	7.25×10^{05}	2.72×10^{05}
*f*18	1.18×10^{06}	3.41×10^{06}	7.30×10^{03}	6.95×10^{08}	7.65×10^{08}	4.27×10^{03}	1.94×10^{08}	1.53×10^{08}	5.34×10^{07}	2.78×10^{04}	2.59×10^{04}	3.68×10^{03}
*f*19	1.95×10^{03}	3.07×10^{01}	1.91×10^{03}	2.12×10^{03}	1.33×10^{02}	1.94×10^{03}	2.00×10^{03}	2.12×10^{01}	1.96×10^{03}	1.93×10^{03}	2.86×10^{01}	1.91×10^{03}
*f*20	3.35×10^{04}	1.40×10^{04}	7.94×10^{03}	3.92×10^{05}	7.12×10^{05}	9.00×10^{03}	5.80×10^{04}	4.45×10^{04}	2.21×10^{04}	8.65×10^{03}	5.61×10^{03}	2.39×10^{03}
*f*21	8.68×10^{05}	1.11×10^{06}	5.19×10^{04}	8.97×10^{06}	9.73×10^{06}	2.76×10^{05}	3.89×10^{06}	2.38×10^{06}	9.50×10^{05}	3.65×10^{05}	3.20×10^{05}	1.02×10^{05}
*f*22	2.87×10^{03}	2.57×10^{02}	2.36×10^{03}	3.35×10^{03}	4.21×10^{02}	2.62×10^{03}	3.33×10^{03}	2.18×10^{02}	2.91×10^{03}	2.68×10^{03}	2.05×10^{02}	2.37×10^{03}
*f*23	2.50×10^{03}	0.00×10^{00}	2.50×10^{03}	2.74×10^{03}	7.28×10^{01}	2.62×10^{03}	2.71×10^{03}	3.52×10^{01}	2.66×10^{03}	2.50×10^{03}	0.00×10^{00}	2.50×10^{03}
*f*24	2.60×10^{03}	0.00×10^{00}	2.60×10^{03}	2.60×10^{03}	3.78×10^{-01}	2.60×10^{03}	2.61×10^{03}	2.56×10^{01}	2.60×10^{03}	2.60×10^{03}	2.08×10^{-06}	2.60×10^{03}
*f*25	2.70×10^{03}	0.00×10^{00}	2.70×10^{03}	2.70×10^{03}	4.68×10^{-01}	2.70×10^{03}	2.74×10^{03}	1.48×10^{01}	2.71×10^{03}	2.70×10^{03}	0.00×10^{00}	2.70×10^{03}
*f*26	2.71×10^{03}	3.03×10^{01}	2.70×10^{03}	2.73×10^{03}	4.35×10^{01}	2.70×10^{03}	2.70×10^{03}	5.37×10^{-01}	2.70×10^{03}	2.70×10^{03}	1.07×10^{-01}	2.70×10^{03}
*f*27	3.48×10^{03}	3.45×10^{02}	3.13×10^{03}	3.64×10^{03}	3.08×10^{02}	3.17×10^{03}	3.99×10^{03}	3.07×10^{01}	3.88×10^{03}	3.38×10^{03}	2.90×10^{02}	3.10×10^{03}
*f*28	4.24×10^{03}	3.70×10^{02}	3.00×10^{03}	4.81×10^{03}	5.02×10^{02}	4.10×10^{03}	3.74×10^{03}	2.34×10^{02}	3.50×10^{03}	4.02×10^{03}	3.10×10^{02}	3.00×10^{03}
*f*29	4.35×10^{06}	6.52×10^{06}	1.54×10^{04}	9.05×10^{06}	6.36×10^{06}	7.04×10^{05}	3.11×10^{03}	1.03×10^{01}	3.11×10^{03}	3.11×10^{03}	2.55×10^{00}	3.11×10^{03}
*f*30	1.67×10^{05}	1.32×10^{05}	1.79×10^{04}	4.00×10^{05}	2.72×10^{05}	5.50×10^{04}	3.74×10^{03}	4.10×10^{02}	3.24×10^{03}	2.18×10^{04}	2.58×10^{04}	3.22×10^{03}
TPIO win	**27**			**29**			**28**			-		

Table 12
Parameter settings of the double-diode model and PV module model.

Parameter	PV module	
	Lower bound	Upper bound
$I_{ph}(A)$	0	2
$I_{01}(\mu A)$	0	50
$I_{02}(\mu A)$	0	50
$R_s(\Omega)$	0	2
$R_{sh}(\Omega)$	0	2000
n_1	1	50
n_2	1	50

Table 13
The experimental results of TPIO and other compared algorithms for the PV module model.

Algorithm	$I_{ph}(A)$	$I_{01}(\mu A)$	$I_{02}(\mu A)$	$R_s(\Omega)$	$R_{sh}(\Omega)$	n_1	n_2	*RMSE*
PSO	1.03141	2.47392	7.97643	1.22905	35.21895	45.4518	49.1336	2.6605×10^{-3}
DA	1.03252	3.51171	3.24156	0.10172	218.94912	41.30324	23.83744	4.8232×10^{-3}
GWO	1.03081	4.92722	2.77663	1.81719	1530.91808	1.79838	40.79118	2.4251×10^{-3}
NPIO	1.03733	4.45441	4.51762	0.18161	149.83028	50	50	2.5478×10^{-3}
PIO	1.03420	3.40894	4.03849	0.01348	132.9821	47.46785	48.96122	2.2053×10^{-3}
SCA	0.93189	4.45517	5.0094	0.87623	1934.84135	50	50	2.1937×10^{-3}
TPIO	1.03091	3.48828	3.60388	0.99192	958.09365	48.2856	47.13631	9.7525×10^{-4}

three unimodal functions, thirteen simple multimodal functions, six mixed functions, and eight Composition Functions. We believe that the optimization goal of all benchmark functions is to minimize. literature [39] gives a detailed definition of the benchmark function as shown in the Table 9. We compared TPIO with six algorithms: PSO, DA, GWO, NPIO, PIO, and SCA. Since the meta-heuristic algorithm has the problem of uncertainty and large randomness in each run, we run each algorithm independently 51 times. The initialization parameters are set as follows: the population size is 120, the dimension is 30, and the maximum evaluation times of the function is 5000. Table 8 shows other setting parameters of the algorithm.

*f*1-*f*3 are three unimodal functions, which are used to evaluate the performance of the convergence speed of the algorithm. They have no local optimum and have only one global optimum position. *f*4-*f*16 are thirteen simple multi-peaked functions, because multi-peaked functions have multiple locally optimal solutions, the purpose of this type of function is to test the ability of the test function to jump out of the local optimum. *f*17-*f*22 are six hybrid functions, and considering the application of the functions in practical optimization problems, different sub-components of the variables may have different properties. In this set of hybrid functions, the variables are randomly divided into a number of sub-function components, and then each sub-function component is optimized using a different basic function. *f*23-*f*30 are eight composition functions, which have a complex structure of multiple locally optimal solutions and multiple globally optimal solutions. The performance of

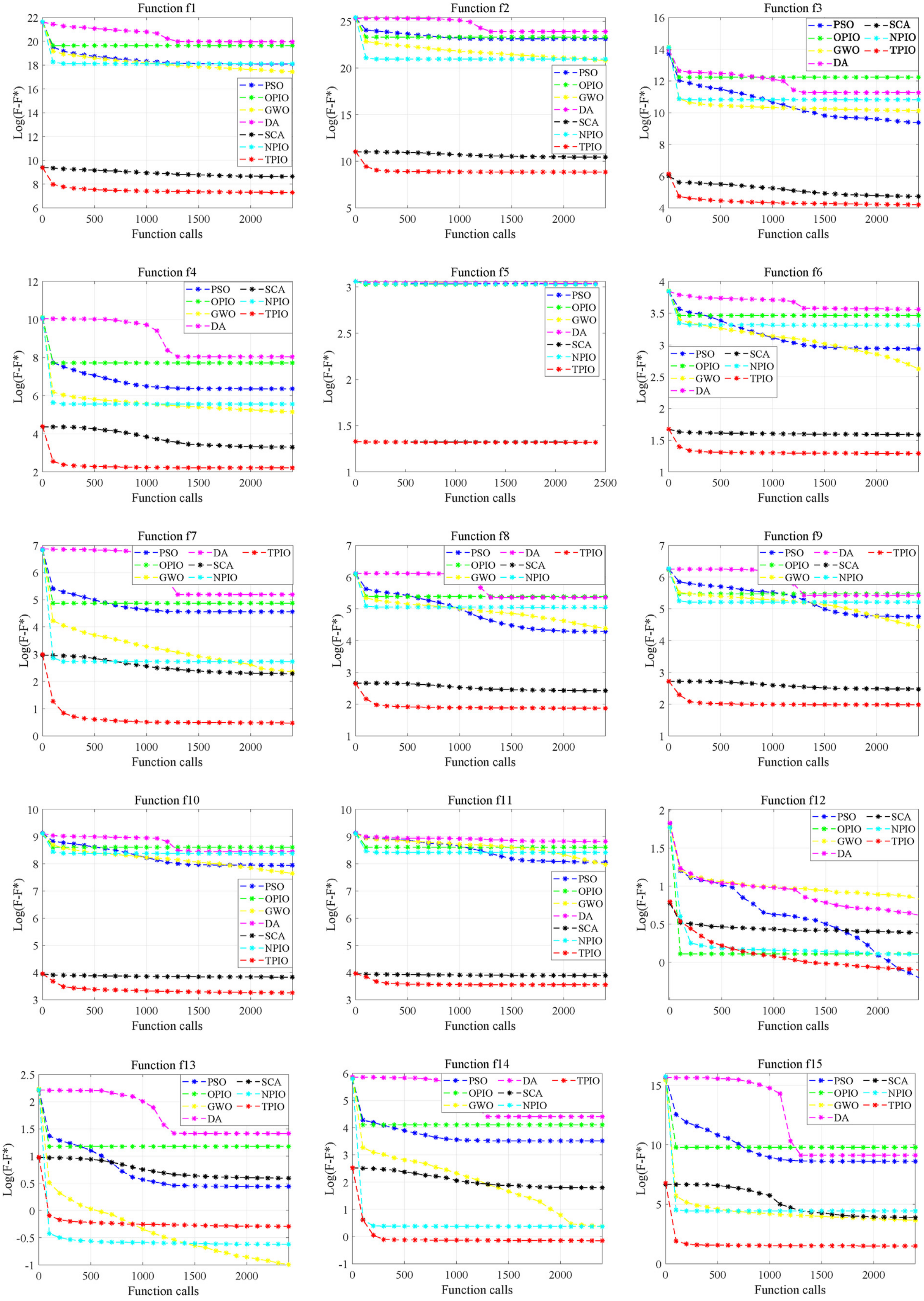


Fig. 6. Convergence curve speed of comparison algorithms on the $f1 - f30$.

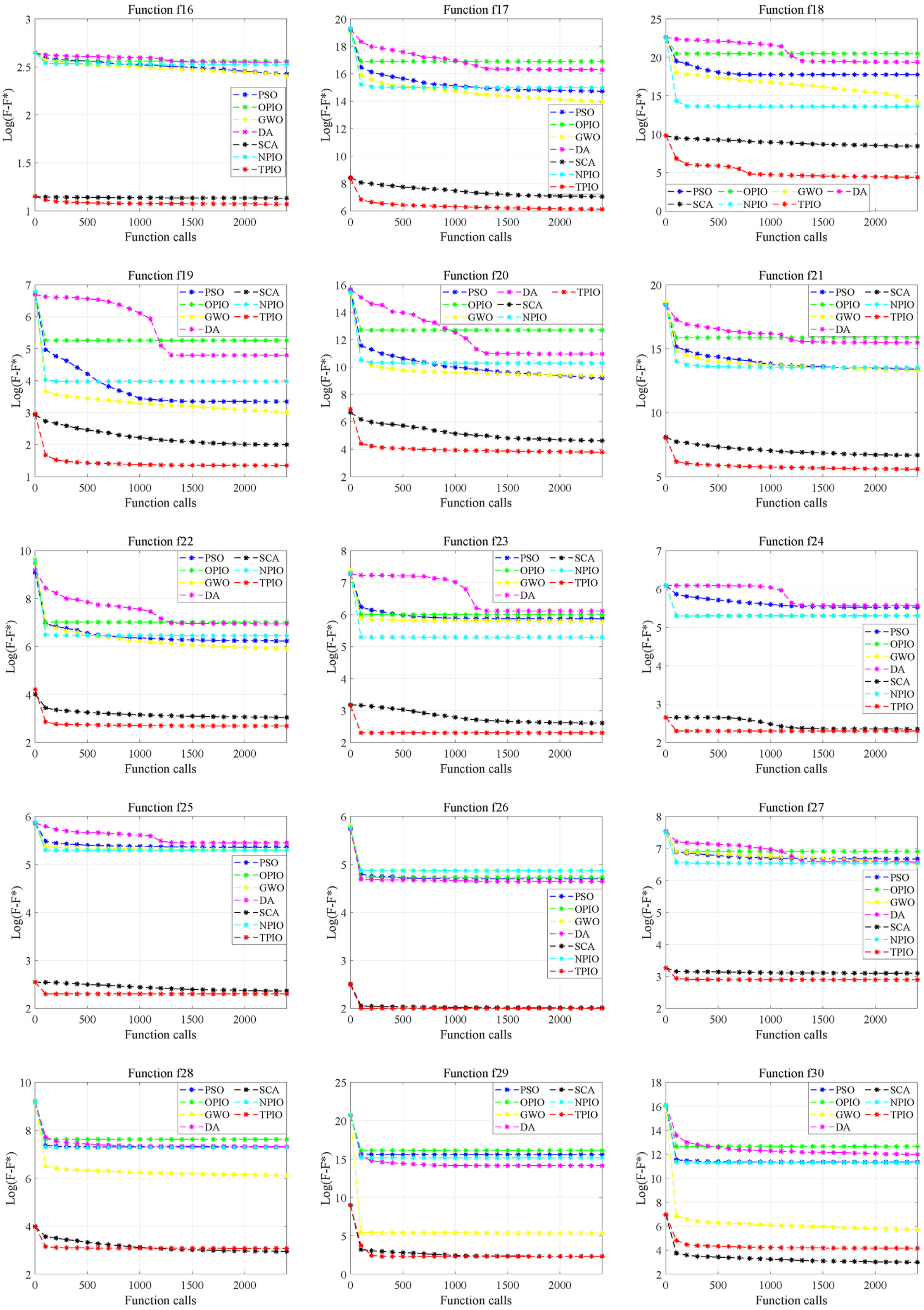


Fig. 6. (*continued*).

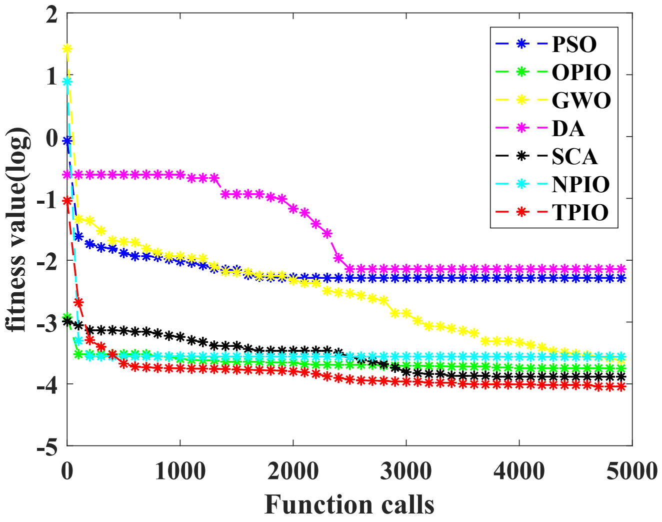


Fig. 7. The convergence curves of the fitness values of PV module model.

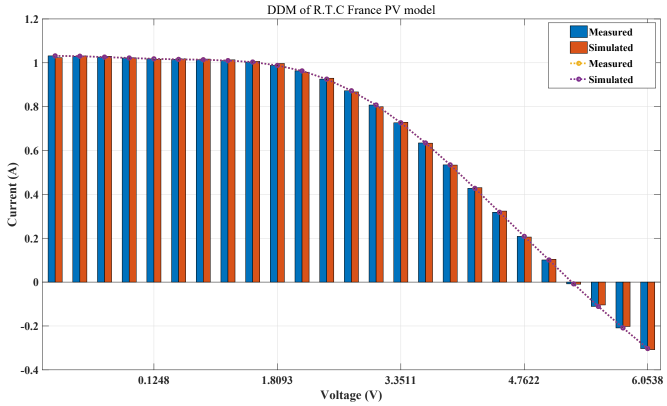


Fig. 8. DDM of RTC France solar cell.

the algorithm in terms of convergence speed, convergence accuracy. it can be basically tested from these 30 test functions in different aspects.

3.2. Applied for parameter identifiable analysis

In this summary, in order to evaluate the performance of the TPIO algorithm in practical applications, we apply the TPIO algorithm to the parameter extraction of the double-diode model and the maximum power tracking of the PV system under cover. The benchmark data set is from the literature, Its data comes from a commercial RTC with a diameter of less than 57 mm at 1000W/m^2 at 33 °C, and a Photowatt-PWP201 photovoltaic module, at 1000 W/m^2 at 33 °C, with 36 polycrystalline silicon cells in series and parallel. The search range of each parameter is shown in Table 12. The algorithm scale of each algorithm is 20, the maximum fitness evaluation times is 2000, and the other parameter settings are the same as those in Table 8. Same as the CEC2014 test function set, each algorithm is run 51 times. Table 13 shows the simulation results of the TPIO algorithm proposed in this paper. Fig. 7 shows the convergence iterative graphs of seven algorithms. It can be seen from the figure that the TPIO algorithm has a fast convergence speed and high convergence accuracy. The search boundary of each parameter is shown in Table 12. After comparison, The initial population size is set to 20, the maximum number of fitness evaluations is set to 5000, and the other parameters are the same as those used in Table 8. For fair comparison, each algorithm has been independently run 51 times in the application of parameter extraction and power tracking.

4. Discussion

Tables 10 and 11 show the comparative experimental data of TPIO and other six algorithms in CEC2014. "*Mean*", "*Std*" and "*Min*" represent the mean variance and minimum value of 51 independent runs. "TPIO win" represents the number of times the algorithm proposed in this article has won compared with the other six algorithms. It can be seen from Tables 10 and 11 that the number of wins between TPIO and PSO, DA, GWO, NPIO, PIO, and SCA are 24, 28, 18, 27, 29, and 28 respectively, which are better than the comparative algorithms, especially Compared with the original PIO algorithm, the performance is very good. However, TPIO is only better than the GWO algorithm 18 times, and only exceeds half of the CEC2014 test function. Fig. 6 shows the convergence iteration rate of the seven algorithms under the test function. The horizontal axis represents the number of evaluations of the algorithm, and the vertical axis represents the decreasing convergence rate of the algorithm.

It can be seen from the Fig. 6 that the TPIO algorithm is used in most test functions. It shows the advantages of fast convergence speed and high convergence accuracy. But in the *f*12 test function, the performance of the TPIO algorithm is not as good as that of the PSO, and the performance of the *f*13 is not as good as the GWO and NPIO. In the *f*30 function, the performance of the TPIO is worse than that of the SCA algorithm. From the above experimental results, it can be proved that the performance of the TPIO algorithm proposed in this paper is better than the compared algorithms on the CEC2014 test function set. The TPIO algorithm reduces the time cost of calculation and discards some insignificant calculation processes. Therefore, the TPIO algorithm is a satisfactory algorithm.

The simulation results of the photovoltaic array model based on the dual diode model using the proposed TPIO algorithm are given. In this model, the TPIO algorithm can get the smallest RMSE value of 9.7525 × 10^{-4}, the worst value is 1.98215 × 10^{-3}, and the average value is 9.98215 × 10^{-4}. Table 13 shows the experimental values of the algorithms compared in this paper. The TPIO algorithm proposed in this paper is significantly better than other algorithms. The advantage of the TPIO algorithm is that it can not only inherit the population reduction trend of the original PIO algorithm, but also ensure that the number of evaluations is more beneficial. In addition, the Taguchi method is cited, which reduces the value of parameters for different dimensions. Based on the above results, it can be analyzed that the TPIO algorithm proposed in this paper has satisfactory results in terms of parameter identification and solar parameter extraction. The comparison value between the calculated current and the measured current after TPIO extracts the parameter value is shown in Fig. 8. According to the above calculation current, the maximum power tracking value under 100% light, 80%, 60%, 40% and 20% is stable at 5.15 × 10^3 W.

5. Conclusions

This paper proposes an algorithm called TPIO, which is a PIO algorithm based on Taguchi method. The TPIO algorithm introduces Taguchi method to strengthen the ability of the PIO algorithm to exploit. The second stage of this algorithm has a mechanism to reduce the population size, which can avoid the evaluation times of some invalid candidates. In addition, the Taguchi method can reduce the number of calculations for different dimensional combinations, increase the randomness of the algorithm, and improve the generation of optimal solutions. The algorithm performance evaluation based on CEC2014 test functions indicated that the convergence speed of TPIO is faster and the convergence accuracy is higher than those of other algorithms. In addition, simulation experiments involving the extraction of internal parameters related to PV systems and the maximum power output were conducted under different illumination conditions, and a dual-diode model was applied. The results show that in terms of solar energy parameter extraction and maximum power tracking. Compared with other algorithms, TPIO can obtain the minimum fitness function value of RMSE is 9.7525×10^{-4}. TPIO outperforms other algorithms based on convergence and accuracy. The main conclusions of this study are as follows:

1. A model study of parameter extraction and maximum power point tracking based on measurement data is proposed.
2. The fitness function value was revised to take into account not only the RMSE of parameter extraction, but also the value of maximum power tracking.
3. The pigeon-inspired optimization algorithm has the problem that it cannot converge well to the optimal position. The Taguchi method can reduce the number of experiments well to calculate. Adding the Taguchi method to the pigeon-inspired optimization algorithm can greatly increase the development capability of the algorithm.
4. Compared with other algorithms, TPIO performs better and more stable on the CEC2014 test function.

The disadvantage of the TPIO algorithm is that although it saves exhaustive calculation time when using Taguchi's method, the setting of dimensions and the construction of orthogonal tables are the focus of the next step in the future research.

Author statement

Jeng-Shyang Pan: Conceptualization, Methodology, Software. **Ai-Qing Tian:** Data curation, Writing Original draft preparation, Writing- Reviewing and Editing. **Václav Snášel:** Visualization, Investigation. **Lingping Kong:** Supervision. **Shu-Chuan Chu:** Software, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Abdelsalam A, Massoud A, Ahmed S, Enjeti P. High-performance adaptive perturb and observe mppt technique for photovoltaic-based microgrids. IEEE Trans Power Electron 2011;26:1010–21. https://doi.org/10.1109/TPEL.2011.2106221.

[2] Al-Atrash H, Batarseh I, Rustom K. Statistical modeling of dsp-based hill-climbing mppt algorithms in noisy environments. In: IEEE applied power electronics conference & exposition; 2005. p. 1773–7.

[3] Al-Atrash H, Batarseh I, Rustom K. Statistical modeling of dsp-based hill-climbing mppt algorithms in noisy environments. In: Twentieth annual IEEE applied power electronics conference and exposition, 2005. APEC 2005; 2005. p. 1773–7. https://doi.org/10.1109/APEC.2005.1453286.

[4] Alajmi BN, Ahmed KH, Finney SJ, Williams BW. Fuzzy-logic-control approach of a modified hill-climbing method for maximum power point in microgrid standalone photovoltaic system. IEEE Trans Power Electron 2011;26:1022–30. https://doi.org/10.1109/TPEL.2010.2090903.

[5] Alireza Askarzadeh, Alireza Rezazadeh. Parameter identification for solar cell models using harmony search-based algorithm. Sol Energy 2012;86:3241–9. https://doi.org/10.1016/j.solener.2012.08.018.

[6] Altas IH, Sharaf AM. A photovoltaic array simulation model for matlab-simulink gui environment. In: 2007 international conference on clean electrical power; 2007. p. 341–5. https://doi.org/10.1109/ICCEP.2007.384234.

[7] Alva G, Lin Y, Fang G. An overview of thermal energy storage systems. Energy 2018;144:341–78.

[8] Awadallah MA. Variations of the bacterial foraging algorithm for the extraction of pv module parameters from nameplate data. Energy Convers Manag 2016;113:312–20. https://doi.org/10.1016/j.enconman.2016.01.071.

[9] Aydin O, Gozde H, D M, C T M. Comparative parameter estimation of single diode pv-cell model by using sine-cosine algorithm and whale optimization algorithm. In: 2019 6th international conference on electrical and electronics engineering (ICEEE); 2019. p. 65–8. https://doi.org/10.1109/ICEEE2019.2019.00020.

[10] Beigi AM, Maroosi A. Parameter identification for solar cells and module using a hybrid firefly and pattern search algorithms. Sol Energy 2018;171:435–46.

[11] Bellia H, Youcef R, Fatima M. A detailed modeling of photovoltaic module using matlab. NRIAG J Astron Geophys 2014;3:53–61. https://doi.org/10.1016/j.nrjag.2014.04.001.

[12] Chu SC, Huang HC, Roddick JF, Pan JS. Overview of algorithms for swarm intelligence. In: Computational collective intelligence. Technologies and applications. Berlin, Heidelberg: Springer Berlin Heidelberg; 2011. p. 28–41.

[13] Chu SC, Tsai PW, Pan JS. Cat swarm optimization. In: Pacific Rim international conference on artificial intelligence. Springer; 2006. p. 854–8.

[14] Ding S, Hipel KW, Dang Yg. Forecasting China's electricity consumption using a new grey prediction model. Energy 2018;149:314–28.

[15] Dorigo M, Di Caro G. Ant colony optimization: a new meta-heuristic. In: Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406). IEEE; 1999. p. 1470–7.

[16] Duan H, Qiao P. Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning. Int J Intell Comput Cybernet 2014. https://doi.org/10.1108/IJICC-02-2014-0005.

[17] Duffie JA, Beckman WA. Solar engineering of thermal processes. A wiley-interscience; 2006.

[18] Enslin Johan HR, Wolf Mario S. Integrated photovoltaic maximum power point tracking converter. IEEE Trans Ind Electron 1997;44:769–73.

[19] Esram T, Chapman PL. Comparison of photovoltaic array maximum power point tracking techniques. IEEE Trans Energy Convers 2007;22:439–49.

[20] Fan Y, Wang P, Heidari AA, Chen H, Mafarja M, et al. Random reselection particle swarm optimization for optimal design of solar photovoltaic modules. Energy 2022;239:121865.

[21] Faris H, Aljarah I, Al-Betar MA, Mirjalili S. Grey wolf optimizer: a review of recent variants and applications. Neural Comput Appl 2018;30:413–35.

[22] Femia N, Granozio D, Petrone G, Spagnuolo G, Vitelli M. Predictive adaptive mppt perturb and observe method. IEEE Trans Aero Electron Syst 2007;43:934–50. https://doi.org/10.1109/TAES.2007.4383584.

[23] Femia N, Petrone G, Spagnuolo G, Vitelli M. Optimization of perturb and observe maximum power point tracking method. IEEE Trans Power Electron 2005;20:963–73.

[24] Ghadimi N, Akbarimajd A, Shayeghi H, Abedinia O. Two stage forecast engine with feature selection technique and improved meta-heuristic algorithm for electricity load forecasting. Energy 2018;161:130–42.

[25] Gomes R, Vitorino MA, Corrêa M, Fernandes DA, Wang R. Shuffled complex evolution on photovoltaic parameter extraction: a comparative analysis. IEEE Trans Sustain Energy 2017.

[26] Hansen K, Breyer C, Lund H. Status and perspectives on 100% renewable energy systems. Energy 2019;175:471–80.

[27] Hua C, Lin J, Shen C. Implementation of a dsp-controlled photovoltaic system with peak power tracking. IEEE Trans Ind Electron 1998;45:99–107.

[28] Jena D, Ramana VV. Modeling of photovoltaic system for uniform and non-uniform irradiance: a critical review. Renew Sustain Energy Rev 2015;52:400–17. https://doi.org/10.1016/j.rser.2015.07.079.

[29] Jiang BQ, Pan JS. A parallel quasi-affine transformation evolution algorithm for global optimization. J Netw Intell 2019;4:30–46.

[30] Kang T, Yao J, Jin M, Yang S, Duong T. A novel improved cuckoo search algorithm for parameter estimation of photovoltaic (pv) models. Energies 2018;11. https://doi.org/10.3390/en11051060.

[31] Karaboga D, Akay B. A comparative study of artificial bee colony algorithm. Appl Math Comput 2009;214:108–32.

[32] Karasu S, Altan A, Bekiros S, Ahmad W. A new forecasting model with wrapper-based feature selection approach using multi-objective optimization technique for chaotic crude oil time series. Energy 2020;212:118750.

[33] Kasa N, Iida T, Liang C. Flyback inverter controlled by sensorless current mppt for photovoltaic power system. IEEE Trans Ind Electron 2005;52:1145–52.

[34] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks. IEEE; 1995. p. 1942–8. https://doi.org/10.1109/ICNN.1995.488968.

[35] King DL. Sandia's pv module electrical performance model. 2000. version, 2000.

[36] King, D.L., Kratochvil, J.A., Boyson, W.E., . Field experience with a new performance characterization procedure for photovoltaic arrays .
[37] Kobayashi K, Takano I, Sawada Y. A study of a two stage maximum power point tracking control of a photovoltaic system under partially shaded insolation conditions. Sol Energy Mater Sol Cell 2006;90:2975–88. https://doi.org/10.1016/j.solmat.2006.06.050.
[38] Laboratories, S.N., . Database of photovoltaic module performance parameters. [EB/OL]. URL: http://www.sandia.gov/pv/docs/Database.htm.
[39] Liang JJ, Qu BY, Suganthan PN. Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization. Comput Intell Lab, Zhengzhou Univ, Zhengzhou China Tech Rep, Nanyang Technol Univ, Singap 2013:490. 635.
[40] Long W, Cai S, Jiao J, Xu M, Wu T. A new hybrid algorithm based on grey wolf optimizer and cuckoo search for parameter extraction of solar photovoltaic models. Energy Convers Manag 2020;203:112243.
[41] Maouhoub N. Photovoltaic module parameter estimation using an analytical approach and least squares method. J Comput Electron 2018;17:784–90. https://doi.org/10.1016/j.apenergy.2017.11.078.
[42] Masafumi Miyatake, Takeshi Inada, Isao Hiratsuka, Hisayo Otsuka, Motomu Nakano. Control characteristics of a fibonacci-search-based maximum power point tracker when a photovoltaic array is partially shaded. In: The 4th international power electronics and motion control conference (IPEMC); 2004. p. 816–21.
[43] Meng Z, Pan JS. Quasi-affine transformation evolution with external archive (quatre-ear): an enhanced structure for differential evolution. Knowl Base Syst 2018;155:35–53.
[44] Mengelkamp E, Gärttner J, Rock K, Kessler S, Orsini L, Weinhardt C. Designing microgrid energy markets: a case study: the brooklyn microgrid. Appl Energy 2018;210:870–80.
[45] Merchaoui M, Sakly A, Mimouni MF. Particle swarm optimisation with adaptive mutation strategy for photovoltaic solar cell/module parameter extraction. Energy Convers Manag 2018;175:151–63. https://doi.org/10.1016/j.enconman.2018.08.081.
[46] Mirjalili S. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Appl 2016;27:1053–73.
[47] Miyatake M, Inada T, Hiratsuka I, Zhao H, Nakano M. Control characteristics of a fibonacci-search-based maximum power point tracker when a photovoltaic array is partially shaded. In: International power electronics & motion control conference; 2003.
[48] Miyatake M, Kouno T, Nakano M. Maximum power point tracking control employing fibonacci search algorithm for photovoltaic power generation system. In: InProceedings of EPE-PEMC; 2001.
[49] Moayedi H, Hayati S. Modelling and optimization of ultimate bearing capacity of strip footing near a slope by soft computing methods. Appl Soft Comput 2018;66:208–19. https://doi.org/10.1016/j.asoc.2018.02.027.
[50] Munkhchuluun E, Meegahapola L, Vahidnia A. Long-term voltage stability with large-scale solar-photovoltaic (pv) generation. Int J Electr Power Energy Syst 2020;117:105663.
[51] Nelson J. The physics of solar cells. The physics of solar cells. by Jenny Nelson. London: Imperial College Press; 2003. https://doi.org/10.1142/p276. 2003 57.
[52] Niu Q, Zhang L, Li K. A biogeography-based optimization algorithm with mutation strategies for model parameter estimation of solar and fuel cells. Energy Convers Manag 2014;86:1173–85. https://doi.org/10.1016/j.enconman.2014.06.026.
[53] Nunes H, Pombo J, Mariano S, Calado M, de Souza JF. A new high performance method for determining the parameters of pv cells and modules based on guaranteed convergence particle swarm optimization. Appl Energy 2018;211:774–91. https://doi.org/10.1016/j.apenergy.2017.11.078.
[54] Ogiela L, Snášel V. Intelligent and semantic threshold schemes for security in cloud computing. Concurrency Comput Pract Ex 2021;33:e5247.
[55] Ojha VK, Abraham A, Snášel V. Metaheuristic design of feedforward neural networks: a review of two decades of research. Eng Appl Artif Intell 2017;60:97–116.
[56] Oliva D, Cuevas E, Pajares G. Parameter identification of solar cells using artificial bee colony optimization. Energy 2014;72:93–102. https://doi.org/10.1016/j.energy.2014.05.011.
[57] Pan JS, Hu P, Chu SC. Binary fish migration optimization for solving unit commitment. Energy 2021:120329.
[58] Pan JS, Tian AQ, Chu SC, Li JB. Improved binary pigeon-inspired optimization and its application for feature selection. Appl Intell 2021:1–19.
[59] Pan JS, Tsai PW, Liao YB. Fish migration optimization based on the fishy biology. In: 2010 fourth international conference on genetic and evolutionary computing. IEEE; 2010. p. 783–6.
[60] Piegari L, Rizzo R. Adaptive perturb and observe algorithm for photovoltaic maximum power point tracking. Renew Power Gener, IET 2010;4:317–28. https://doi.org/10.1049/iet-rpg.2009.0006.
[61] Qiao W, Moayedi H, Foong LK. Nature-inspired hybrid techniques of iwo, da, es, ga, and ica, validated through a k-fold validation process predicting monthly natural gas consumption. Energy Build 2020;217:110023. https://doi.org/10.1016/j.enbuild.2020.110023.
[62] Rasheed MS, Shihab S. Modelling and parameter extraction of pv cell using single-diode model. Adv Energy Conv Mater 2020:96–104.
[63] Ridha HM, Heidari AA, Wang M, Chen H. Boosted mutation-based harris hawks optimizer for parameters identification of single-diode solar cell models. Energy Convers Manag 2020;209:112660.
[64] Safari A, Mekhilef S. Simulation and hardware implementation of incremental conductance mppt with direct control method using cuk converter. IEEE Trans Ind Electron 2011;58:1154–61. https://doi.org/10.1109/TIE.2010.2048834.
[65] Senturk A. Investigation of datasheet provided temperature coefficients of photovoltaic modules under various sky profiles at the field by applying a new validation procedure. Renew Energy 2020;152:644–52.
[66] Shayeghi H, Mahdavi M, Bagheri A. An improved dpso with mutation based on similarity algorithm for optimization of transmission lines loading. Energy Convers Manag 2010;51:2715–23. https://doi.org/10.1016/j.enconman.2010.06.007.
[67] Snasel V, Kong L, Tsai PW, Pan JS. Sink node placement strategies based on cat swarm optimization algorithm. J Netw Intell 2016;1:52–60.
[68] Sörensen K. Metaheuristics—the metaphor exposed. Int Trans Oper Res 2015;22:3–18. https://doi.org/10.1111/itor.12001.
[69] Sweidan AH, El-Bendary N, Hegazy OM, Hassanien AE, Snasel V. Water pollution detection system based on fish gills as a biomarker. Procedia Comput Sci 2015;65:601–11.
[70] Tey KS, Mekhilef S, Safari A. Simple and low cost incremental conductance maximum power point tracking using buck-boost converter. J Renew Sustain Energy 2013;5. https://doi.org/10.1063/1.4794749.
[71] Tian AQ, Chu SC, Pan JS, Cui H, Zheng WM. A compact pigeon-inspired optimization for maximum short-term generation mode in cascade hydroelectric power station. Sustainability 2020;12.
[72] Vantuch T, Mišák S, Ježowicz T, Buriánek T, Snášel V. The power quality forecasting model for off-grid system supported by multiobjective optimization. IEEE Trans Ind Electron 2017;64:9507–16.
[73] Veerachary Mummadi, Senjyu Tomonobu, Uezato Katsumi. Neural-network-based maximum-power-point tracking of coupled-inductor interleaved-boost-converter-supplied pv system using fuzzy controller. IEEE Trans Ind Electron 2003;50:749–58.
[74] Wu Z, Yu D, Kang X. Parameter identification of photovoltaic cell model based on improved ant lion optimizer. Energy Convers Manag 2017;151:107–15. https://doi.org/10.1016/j.enconman.2017.08.088.
[75] Xiao W, Dunford WG. A modified adaptive hill climbing mppt method for photovoltaic power systems. In: 2004 IEEE 35th annual power electronics specialists conference (IEEE cat. No.04CH37551); 2004. p. 1022–30.
[76] Xiong G, Zhang J, Shi D, Zhu L, Yuan X, Tan Z. Winner-leading competitive swarm optimizer with dynamic Gaussian mutation for parameter extraction of solar photovoltaic models. Energy Convers Manag 2020;206:112450.
[77] Yu K, Liang JJ, Qu BY, Chen X, Wang H. Parameters identification of photovoltaic models using an improved jaya optimization algorithm. Energy Convers Manag 2017;150:742–53.
[78] Zeng H, Liu X, W W, X S. New results on stability analysis of systems with time-varying delays using a generalized free-matrix-based inequality. J Franklin Inst 2019;356:7312–21. https://doi.org/10.1016/j.jfranklin.2019.03.029.
[79] Zhang C, Allafi W, Dinh Q, Ascencio P, Marco J. Online estimation of battery equivalent circuit model parameters and state of charge using decoupled least squares technique. Energy 2018;142:678–88.
[80] Zhang L, Hurley W, Wolfle W. A new approach to achieve maximum power point tracking for pv system with a variable inductor. 2010. p. 948–52. https://doi.org/10.1109/PEDG.2010.5545758.
[81] Zhao Y, Ma K, Bai W, Du D, Zhu Z, Wang Y, Gao J. Energy-saving thermally coupled ternary extractive distillation process by combining with mixed entrainer for separating ternary mixture containing bioethanol. Energy 2018;148:296–308.

Jeng-Shyang Pan received the B.S. degree in electronic engineering from the National Taiwan University of Science and Technology in 1986, the M.S. degree in communication engineering from National Chiao Tung University, Taiwan, in 1988, and the Ph.D. degree in electrical engineering from the University of Edinburgh, U.K., in 1996. He is currently the Professor in Shandong University of Science and Technology. He is also the Professor with the Harbin Institute of Technology. He is the IET Fellow, U.K., and has been the Vice Chair of the IEEE Tainan Section. His current research intrests include the computational intelligence, smart grid and wireless sensor network.

Ai-Qing Tian received his B.S. degree from Taishan College of Scienceand Technology, Taian, China, in 2019. He is currently pursuing the master degree with the Shandong University of Science and Technology, Qingdao, China. His recent research interests are swarm intelligence and artificial neural networks.

Václav Snášel (Senior Member, IEEE) received a master's degree in numerical mathematics from the Faculty of Science, Palacky University, Olomouc, Czech Republic, in 1981, and the Ph.D. degree in algebra and number theory from Masaryk University, Brno, Czech Republic, in 1991. He is currently a Full Professor with the VŠB - Technical University of Ostrava, Czech Republic. His research and development experience includes more than 30 years in the industry and academia. He works in a multidisciplinary environment involving artificial intelligence, social networks, conceptual lattice, information retrieval, semantic web, knowledge management, data compression, machine intelligence, and nature and bioinspired computing applied to various real-world problems. He has authored or co-authored several refereed journal/conference papers, books, and book chapters. Prof. Snášel is the Chair of the IEEE International Conference on Systems, Man, and Cybernetics, Czechoslovak Chapter. He also served as an Editor/Guest Editor for several journals, such as Engineering Applications of Artificial Intelligence (Elsevier), Neurocomputing (Elsevier), and Journal of Applied Logic (Elsevier).

Lingping Kong received a master's degree and Ph.D. degree in computer applied technology, Harbin institute of technogy, Shenzhen, China, in 2013, and 2018. She is currently studying at VŠB - Technical University of Ostrava, Czech Republic. Her research interests include multi-objective optimization and its applications.

Shu-Chuan Chu received the Ph.D. degree in 2004 from the School of Computer Science, Engineering and Mathematics, Flinders University of South Australia. She joined Flinders University in December 2009 after 9 years at the Cheng Shiu University, Taiwan. She is the Research Fellow in the College of Science and Engineering of Flinders University, Australia from December 2009. Currently, she is the Research Fellow with PhD advisor in the College of Computer Science and Engineering of Shandong University of Science and Technology from September 2019. Her research interests are mainly in Swarm Intelligence, Intelligent Computing and Data Mining.