An improved random drift particle swarm optimization-feed forward backpropagation neural network for high-precision state-of-charge estimation of lithium-ion batteries.

HAI, N., WANG, S., LIU, D., GAO, H. and FERNANDEZ, C.

2023

An improved random drift particle swarm optimization-feed forward backpropagation neural network for high-precision state-of-charge estimation of lithium-ion batteries

Nan Hai [a], Shunli Wang [a,*], Donglei Liu [a], Haiying Gao [a], Carlos Fernandez [b]

[a] School of Information Engineering, Engineering and Technology Center, Southwest University of Science and Technology, Mianyang 621010, China
[b] School of Pharmacy and Life Sciences, Robert Gordon University, Aberdeen AB10-7GJ, UK

* Corresponding author - *mail address:* wangshunli@swust.edu.cn (S. Wang).

A B S T R A C T

A predictive model with high accuracy and stability of the state of charge (SOC) estimation for lithium-ion batteries plays a significant role in electric vehicles. An improved random drift particle swarm optimization- feed forward backpropagation neural network (IRDPSO-FFBPNN) is established in this paper. Basically, a three-layer FFBPNN is established, and its learning process is analyzed in detail. Then, to avoid the particle out- of-control, inducting weight parameter σ to achieve dynamic control weight convergence. What's more, the cross-reorganization of data is proposed to enhance the utilization. Finally, a further performance comparison with other networks is made under different working conditions to prove the effectiveness of the IRDPSO- FFBPNN. The experimental results showed that the maximum SOC error of the IRDPSO-FFBPNN is 0.1021% in 45s, 0.1237% in 116s under BBDST and DST with different temperatures, respectively, which performed better both in terms of time-consumption and accuracy.

Keywords: Backpropagation; Random particle swarm; Dynamic weight adjustment; Cross-reorganization; State of charge; Lithium-ion batteries

1. Introduction

With the increase of awareness about environmental protection, sustainable and efficient applications have become a common goal in the world [1], lithium batteries are widely used in common facilities as a specific representative with their many advantages including fast charge [2], long lifespan, no memory effect, lightweight, etc. [3], and electric vehicles play a major role in applications with lithium-ion batteries. Thus, a system of battery management to monitor the status in real-time is significant [4], and the SOC is the basis of the safe and stable operation of lithium-ion batteries [5], while it cannot be measured directly [6]. Parameters and methods will help to achieve the estimation for its nonlinear system [7].

For SOC estimation, the methods of direct estimation, filter models, and data-driven are more developed [8]. Direct methods include the Open-circuit voltage (OCV) method and the Ampere-hour (Ah) integral method [9], in which many assumptions and empirical parameters are applied, and the solution of equation system is complicated [10], so a combination of OCV and coulomb counting methods (OCV-CC) is proposed to measure the variable temperatures of batteries [11]. As for the filtering model, a new adaptive extended Kalman filter (AEKF) algorithm is designed [12], and it has superior performance than the conventional extended Kalman Filter (CEKF) under dynamic load conditions. Data-driven becomes mainstream with their simple structures [13] as deep learning evolves, and they can close to random complex nonlinear mapping with good performance to establish a high accuracy system [14]. An improved long short-term memory recurrent neural network (LSTM-RNN) with extended input (EI) and constrained output (CO) [15], which performed high stability of the system.

Different optimization algorithms have different focuses and different effects [16]. In the 1970s, the GA is first proposed by John Holland in the USA with the process of selection, cross, and mutation, which imitated from the adaptive change mechanisms of genes in nature to generate a new population [17–19]. Different from the GA, the differential evolution (DE) algorithm implements population perturbations by selecting particle differential information randomly, which is currently one of the most powerful swarm intelligence optimization algorithms [20–22]. The PSO algorithm synthesizes self-experience and group information when foraging by simulating the ability of birds so that the particles in the algorithm can show similar search behavior to birds foraging [23]. In 2013, Sun et al. proposed the RDPSO algorithm [24], which was designed from the motion model of free electrons in metal conductors. The movement of particles attracted by the individual best (ibest) position they found and attracted by the global best (gbest) location found by the entire population [25]. Other common neural networks used for SOC prediction of lithium-ion batteries include con-volutional neural networks (CNN), fully connected neural networks (FCNN), and backpropagation neural network (BPNN).

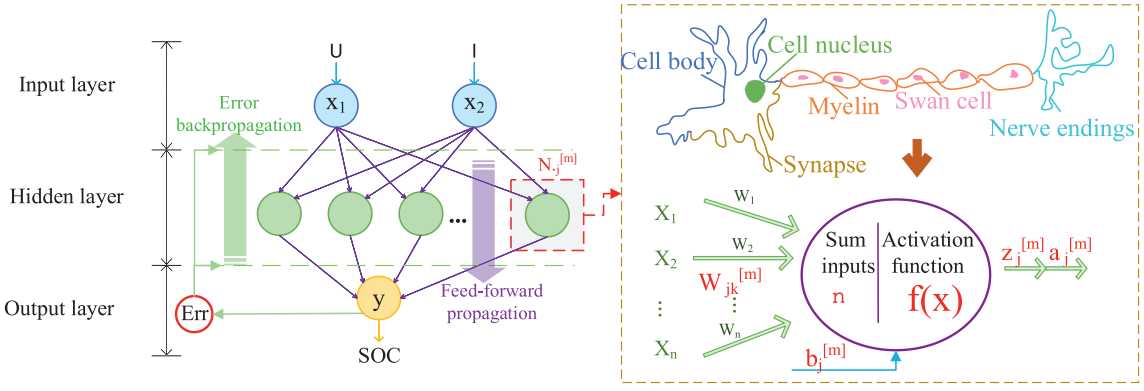


Fig. 1. The structure of FFBPNN and its neuron.

Table 1
Specific steps of error backpropagation.

Confirming the output of the hidden layer and the output layer nodes, as shown in Eq. (5).

Step 1.
$$\begin{cases} H_j(n) = f\left(\sum_{i=0}^{n} W_{ji}^{1} x_i - \theta_j\right) \\ Y_k(n) = f\left(\sum_{i=0}^{n} W_{kj}^{2} x_i - \theta_k\right) \end{cases} \quad (5)$$

Calculating the Err by the Eq. (6). According to the neural network model established in Fig. 1, the values of m can be determined as 1 and 2, representing the input layer to the hidden layer and the hidden layer to the output layer, respectively.

Step 2.
$$\begin{cases} E_j^{[m]} = \frac{1}{2}\sum_{k=1}^{n} e_j^2 \\ E = \sum_{j=1}^{N} E_j = \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{n} e_j^2 \\ e_j = y_k - \hat{y}_{jk} \end{cases} \quad (6)$$

Calculation of the error function and the partial derivatives of the output layer and the hidden layer nodes are shown in Eq. (7).

Step 3.
$$\begin{cases} \frac{\partial E}{\partial W_{kj}^{2}} = \sum_{k=1}^{l} \frac{\partial E^{*}}{\partial Y_k} \frac{\partial Y_k}{\partial W_{kj}^{2}} = \frac{\partial E^{*}}{\partial Y_k} \frac{\partial Y_k}{\partial W_{kj}^{2}} = -e_k^{*} Y_k^{'} * Y_j \\ \frac{\partial E}{\partial W_{ij}^{1}} = \sum_{i}\sum_{j} \frac{\partial E^{*}}{\partial Y_k} \frac{\partial Y_k^{*}}{\partial H_j} \frac{\partial H_j}{\partial W_{ij}^{1}} = -Y_j^{'} * \sum_{k} e_k^{*} Y_k^{'} W_{jk}^{2} \end{cases} \quad (7)$$

Step 4. Using the gradient descent principle, let the connection weights be corrected according to the learning rate σ. The connection weights of the nodes in the output and hidden layers are updated into Eq. (8).
$$\begin{cases} W_{jk}^{2}(m+1) = W_{jk}^{2}(m) + \Delta W_{kj}^{2} = W_{kj}^{2}(m) + \sigma e_k^{*} Y_k^{'} H_j \\ W_{ji}^{1}(m+1) = W_{ji}^{1}(m) + \Delta W_{ji}^{1} = W_{ji}^{2}(m) + \sigma^{'} H_j^{'} * e_k^{*} Y_k^{'} W_{kj}^{2}(m) * x \end{cases} \quad (8)$$

Step 5. Let m = m + 1, until the error is less than the expected value, the neural network learning is finished.

In terms of the topology of BPNN, it is highly dependent on initial weights and thresholds [26]. An artificial fish swarm algorithm-BPNN (AFSA-BPNN) can verify that it is more realistic than the EKF algorithm with a maximum error of 0.5 % [27]. What's more, Levy's flight strategy (LPSO) algorithm based on BPNN (LPSO-BPNN) is proposed to optimize the weights and thresholds [28], which is carried out by using NASA charging and discharging data, while the method is not verified under complex working conditions. Then, a fractional order model with the BPNN algorithm was adopted [29], and its experiments with the NCR18650B lithium-ion battery were made at different temperatures, which limits the maximum error to 10 %. Correspondingly, the combination of neural networks and filter models is often used to achieve SOC and state of health (SOH) estimation of lithium-ion batteries [30]. A novel BPNN-dual extended Kalman filter (BPNN-DEKF) method for SOC and SOH co-estimation based on limited memory recursive least square algorithm is proposed by establishing a second-order equivalent circuit model [31], which offered a high accuracy and robustness method in comparison with single DEKF algorithm at 2.03 % of maximum error [32]. An ant lion optimizer based on BPNN (ALO-BPNN) and UKF method is proposed to predict the SOC with high accuracy [33]. And the robustness and stability of neural networks need to be verified under complex operating conditions and at different temperatures [34]. In this paper, the IRDPSO-FFBPNN is established. Particularly, the weight error parameter is induced to control the step of the particle to achieve dynamic convergence with trajectory drift. Then, the method of cross-reorganization is established to improve dataset utilization. What's more, the performance of the proposed methods is verified under the DST, and BBDST working conditions independently, then the accuracy and stability of the IRDPSO-FFBPNN are analyzed in detail using mean absolute error (MAE), and the goodness of fit (R2), root mean square error of prediction (RMSEP), comparing its performance with other existing algorithms. In conclusion, the paper conducts three parts: establishing an accurate RDPSO-FFBPNN model, proposing two methods to improve the accuracy and efficiency of the system, and verifying the performance of the IRDPSO-FFBPNN under different working conditions.

2. Improved RDPSO-FFBPNN modeling and mathematical analysis

2.1. FFBPNN establishment

The FFBPNN has two processes including feed forward propagation and error backpropagation [35]. In general, the forward propagation completes transmitting the sample data from the input layer to the output layer. If the output cannot meet the conditions, the erroneous value will be traversed back [36]. The typical multi-layer perception network is a three-layer hierarchical neural network, including the input layer, the hidden layer, and the output layer [37]. According to pre-diction content of lithium-ion batteries, the current and voltage are set as input layers, and the SOC is set as the output layer, in this way, a three-layer FFBPNN structure can be confirmed as 2–5-1 according to the Eq. (1).

$$H = \sqrt{m+n} + \alpha \quad (1)$$

H represents the number of the hidden layer, *n* means the number of the input layer, *m* is the number of the output layer, *α* is an interval variable from 1 to 10, when the value of α is 3, the request is satisfied with high matching. The network edges connect the processing units called neurons. Neurons mimic the principle and function of biological neurons to complete the weighting, summing, and transferring of data, and the structure of network and a certain neuron is shown in Fig. 1.

(1) Feed-forward propagation

In Fig. 1, a random neuron $N_j^{[m]}$ is the *j*th neuron from *m*th layer with *n* inputs is selected to analyze. The input is $(x_1, x_2, x_3, x_4, \ldots x_n)^T$, its corresponding variable weight matrix is $(w_1, w_2, w_3, w_4, \ldots w_n)^T$ and the $b_j^{[m]}$ represents the deviation, then the linear input in summation is $z_j^{[m]} = w_1x_1 + w_2x_2 + w_3x_3 + \ldots + w_nx_n + b_j^{[m]}$, and $y_j^{[m]} = f(*)$, and the *f(*)* is an activation function [23]. The input $z_j^{[m]}$ of the network can be analyzed as shown in Eq. (2).

$$Z_j^{[m]} = \sum_{i=1}^{n} w_{ji}^{[m]} x_{ji}^{[m]} + b_j^{[m]} = W_j X + b_j \tag{2}$$

Then the output of the system can be observed in Eq. (3) [38].

$$y_j = f(Z_j) = f\left(\sum_{i=i}^{n} w_{ji}^{[m]*} x_{ji}^{[m]} \right) = F(W_j X) \tag{3}$$

The sigmoid function and tanh function are two common S-type saturation functions that are widely used as activation functions in neural networks [39]. The sigmoid function is smooth and easy to derive, and its derivative function with respect to *x* can be expressed in terms of itself. The sigmoid function can meet the activation conditions and the expression is elucidated in Eq. (4).

$$f(x) = \frac{1}{1 + e^{-x}} \tag{4}$$

(2) Error backpropagation

Gradient descent is an implementation of backward propagation of error that cause the weight of each training sample to vary along a negative gradient until *E* is minimized [26]. According to the network structure in Fig. 1, the nodes of the input layer, the implicit layer and the output layer are set as *Xi(n), Hj(n), Yk(n)*, the connection weight of the ith neuron in the input layer to the *jth* neuron in the hidden layer is W^1_{ji}, similarly, the connection weight of the jth neuron in the hidden layer to the kth neuron in the output layer is W^2_{kj}. The specific steps of error backpropagation are in Table 1.

2.2. RDPSO algorithm

The RDPSO algorithm is a PSO variant inspired by the free election model in metal conductors placed in an external electric field [40], and the search of each particle is regarded as the same as the free electron in the metal inductors. Basically, each of the *M* particles is treated as a volume-free individual in *N*-dimensional space, with the current position and velocity vector of the *i*th particle at the *m*th iterations elaborated as $X_i^m = (X_i^1, X_i^2, X_i^3, \ldots X_i^N)$ and $V_i^m = (V_i^1, V_i^2, V_i^3, \ldots V_i^N)$, Each particle *i* also has the vector of its personal best (pbest) position $P_{ibest} = X_i^m = (P_i^1, P_i^2, P_i^3 \ldots P_i^N)$. Besides, there is a vector $P_{gbest} = G_n = (G_n^1, G_n^2, G_n^3, \ldots G_n^N)$, known as global best (gbest) position. Accordingly, the vector of position and velocity can be updated by Eq. (9).

$$V_{i,n+1}{}^j = w_p V_{i,n}{}^j + c_1 r_{i,n}{}^j \left(p_{i,n}{}^j - X_{i,n}{}^j \right) + \left(c_2 R_{i,n}{}^j \left(G_n{}^j - X_{i,n}{}^j \right), r_{i,n}^j \sim U(0,1), R_{i,n}^j \sim U(0,1) \right) \tag{9}$$

Similarly, the position will be updated by Eq. (10).

$$X_{i,n+1}{}^j = X_{i,n}{}^j + V_{i,n+1}{}^j \tag{10}$$

For $i = 1,2,3,\ldots Q, j = 1,2,3 \ldots Z$, and the parameter c_1 and c_2 are the acceleration factors, then the $P_{i,n+1}$ can be found by

$$P_{i,n+1} = \begin{cases} P_{i,n}, f(P_{i,n}) < f(X_{i,n+1}) \\ X_{i,n+1}, f(P_{i,n}) \geq f(X_{i,n+1}) \end{cases} \tag{11}$$

$$P_{ibest} = \frac{c_1 r_{i,n}{}^j p_{ibest}{}^j + c_2 R_{i,n}{}^j P_{gbest}{}^j}{c_1 r_{i,n}{}^j + c_2 R_{i,n}{}^j} \tag{12}$$

where the *f(*)* means the target function of its corresponding position fitness. The convergence speed is limited for the search range is too large, the behavior of the particle is considered a superposition of thermal and drift motion in the RDPSO algorithm, and the velocity of each particle has two components including random and drift part as shown in Eq. (13).

$$V_{i,n-1}{}^k = VR_{i,n-1}{}^k + VD_{i,n-1}{}^k \tag{13}$$

In Eq. (13), $VR^k_{i,n-1}$ and $VD^k_{i,n-1}$ represent the random component and drift component of particle *i* in its *j* dimension of *n-1*th iterations. Before the analysis, the flow chart is shown in Fig. 2.

(1) The analysis of the random component $VR_{i,n-1}^{k}$.

The random component $VR_{i,n-1}^{k}$ achieves the global search to replace the partial function of inertia weight in the traditional algorithm, and the probability density is shown in Eq. (14).

$$f_{v,R_{i,n-1}{}^{k}}(v) = \frac{1}{\sqrt{2\pi\delta_{i,n-1}{}^{k}}} e \frac{-v^2}{2\left(\delta_{i,n-1}{}^{k}\right)^2} \tag{14}$$

In Eq. (14), the $\delta_{i,n-1}^{k}$ means the standard deviation of the Gaussian distribution [41], and the Maxwell is supposed to support the random component $VR_{i,n-1}^{k}$, which obeys the Gaussian distribution. Then, the expression can be exchanged as shown in Eq. (15).

$$\begin{cases} VR_{i,n-1}{}^{k} = \delta_{i,n-1}{}^{k}\mu_{i,n-1}{}^{k}, \mu \sim [0,1] \\ \delta_{i,n-1}{}^{k} = \varepsilon |Ave_{n-1}{}^{k} - X_{i,n-1}{}^{k}| \\ Ave_{i,n-1}{}^{k} = \dfrac{1}{N}\sum_{i=1}^{N} P_{ibest,n-1}{}^{k} \end{cases} \tag{15}$$

In Eq. (15), the parameter ε is the expansion coefficient, the Ave_{n-1}^{k} means the average of current optimal positions P_{ibest}, and based on the definition of average. The N means the population size, it can be updated as shown in Eq. (16).

$$VR_{i,n-1}{}^{k} = \varepsilon |Ave_{i,n-1}{}^{k} - X_{i,n-1}{}^{k}| \mu_{i,n-1}{}^{k} \tag{16}$$

(2) The analysis of the drift component $VD_{i,n-1}^{k}$.

The drift component $VD_{i,n-1}^{k}$ aims at achieving local search of system, and the initial expression of traditional PSO algorithm can be inducted to the RDPSO algorithm that can be observed in Eq. (17).

$$VD_{i,n-1}{}^{k} = c_1 r_{i,n-1}{}^{k}\left(p_{ibest,n-1}{}^{k} - X_{i,n-1}{}^{k}\right) + c_2 R_{i,n-1}{}^{k}\left(p_{gbest,n-1}{}^{k} - X_{i,n-1}{}^{k}\right) \tag{17}$$

And the as definition, the coefficient $c_1 r_{i,n-1}^{k}$ and $c_2 R_{i,n-1}^{k}$ is the com-bination of random value, and from the perspective of the definition of P_{ibest}, it needs to induct a linear expression to limit the randomization of particle motion, which is shown in Eq. (18).

$$VD_{i,n-1}{}^{k} = \lambda\left(p_{ibest,n-1}{}^{k} - X_{i,n-1}{}^{k}\right) \tag{18}$$

The λ is the drift coefficient, which is limited at interval [0,2], and the particles tend to close to the position of the P_{ibest}. The expression is updated as shown in Eq. (19).

$$V_{i,n}{}^{k} = \varepsilon |Ave_{i,n-1}{}^{k} - X_{i,n-1}{}^{k}| \delta_{i,n-1}{}^{k} + \lambda\left(p_{ibest,n-1}{}^{k} - X_{i,n-1}{}^{k}\right) \tag{19}$$

In the RDPSO algorithm, $\varepsilon$and $\lambda$are two significant user-specified parameters that can be adjusted to balance the local and global search of the particles [42]. The procedure of the algorithm is outlined below in Table 2.

2.3. Improved RDPSO-FFBPNN estimation strategy

The RDPSO algorithm uses the initialized particles to optimize the weights and thresholds of the BPNN and activates the error of the network as a fitness function of the RDPSO algorithm, which is continuously corrected through the FFBPNN error feedback mechanism to achieve the set target. Introducing directional parameters to control the scale of particle motion, while reinforcing network learning through cross-reorganization The flow chart of the IRDPSO-FFBPNN is shown in Fig. 3.

In Fig. 3, the *MSE* is set as the fitness input to train the initial structure, which is like the backpropagation process in the FFBPNN, it sends the error of the system back to the hidden layer and the input layer, the parameter that chosen as the value to input the fitness can be various including *MAE*, *MSE*, R^2, *RMSEP*, *MAPE*.

(1) Induct the weight error σ——achieve the dynamic convergence of weights

The parameter σ is inducted to control the inertia weight randomization, which can avoid the particle miss the P_{gbest} in the process of closing. The particle is stable to tend to the individual optimal position, but it cannot be proved that the process tends to converge in the process. The definition of particle motion convergence and the random optimization algorithm is different, for the process of update from the individual particle is random, and it is difficult to analyze all particles when they transmit the information about the position and velocity to each other, and it is corresponding with the Pibest and Pgbest . The randomness of the RDPSO algorithm will help particles escape the local optimization, but in the process of iterations, the phenomenon appeared that the particle tends to cross the global optimal (x2, y2) to reach another local optimal position Pibest and it is marked as point (x3,y3) which illustrates the particle misses the metrics to close to the global optimal Pgbest , especially the Pibest and the Pgbest are on the same slope of the directive motion of a certain particle. The illustration with figure format is shown in Fig. 4.

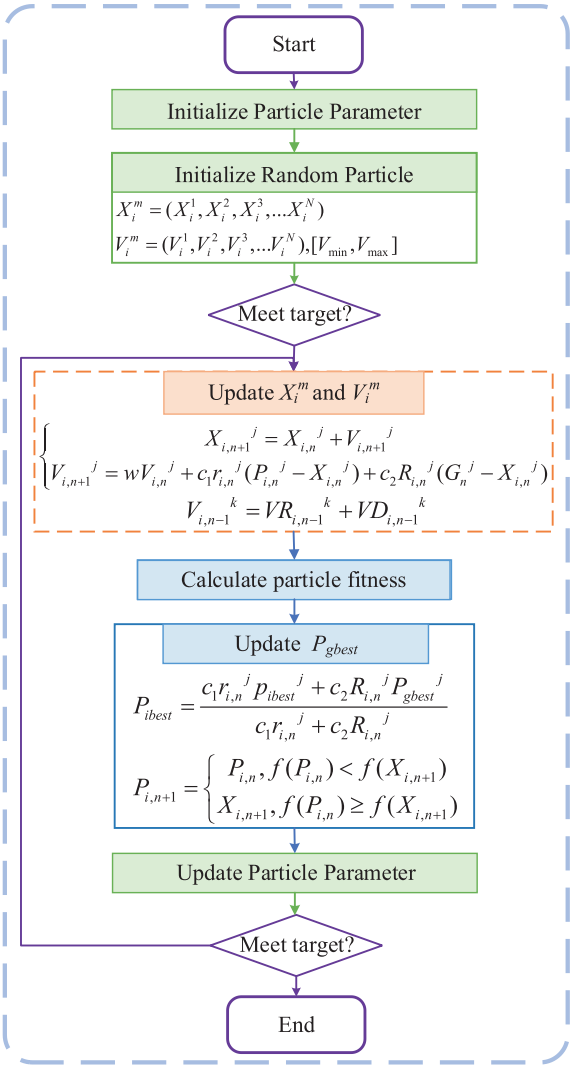


Fig. 2. The flow chart of RDPSO.

Table 2
The pseudocode of RDPSO algorithm.

Start

Set the P_{ibest} to be the current position of each particle;

Set n=n+1;

Compute the $Ave_{i,n-1}^{k}$ and select the value of ε and λ.

for i= 1 to N do

for k= 1 to M do

$p_{i,n-1}^{k} = \delta_{i,n-1}^{\ k} P_{i,n-1}^{k} + P_{i,gbest}^{k} - \delta_{i,n-1}^{\ k}$

$V_{i,n}^{\ k} = \varepsilon \,|\, Ave_{i,n-1}^{\ k} - X_{i,n-1}^{\ k} \,|\, \delta_{i,n-1}^{\ k} + \lambda(p_{ibest,n-1}^{\ k} - X_{i,n-1}^{\ k})$

End

Evaluate the fitness.

Update the $P_{ibest,n-1}^{k}$ and $P_{gbest,n-1}^{k}$.

end

end

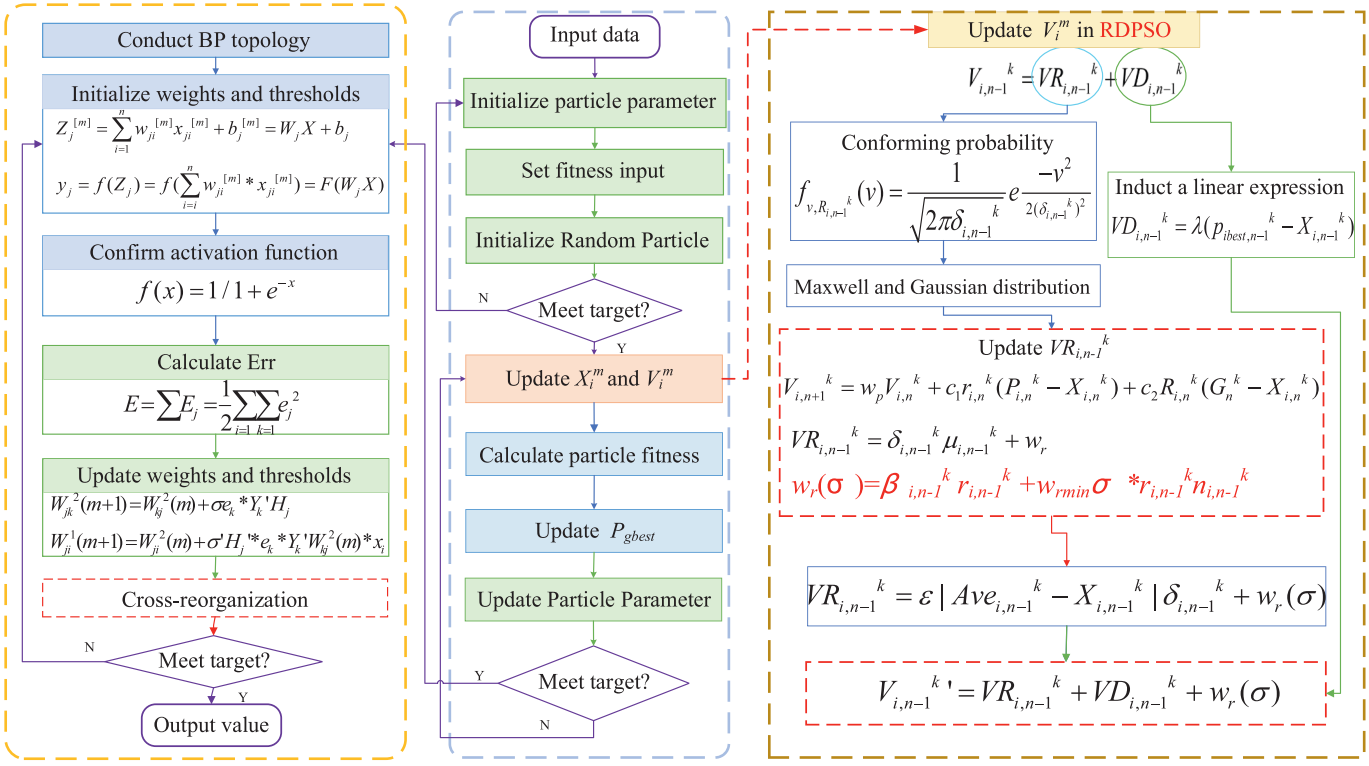


Fig. 3. The flow chart of the IRDPSO-FFBPNN.

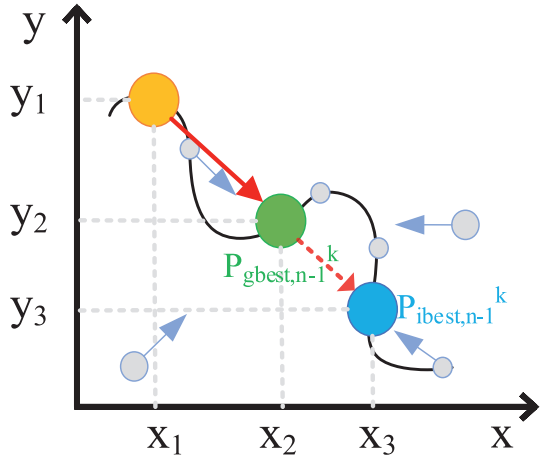


Fig. 4. The random motion deviant trajectory of a certain particle.

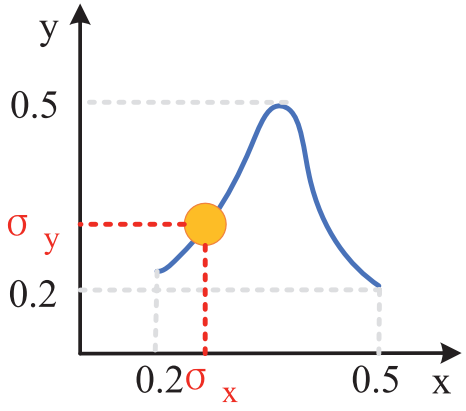


Fig. 5. Adjust motion with σ at the interval.

Table 3
Pseudocode for cross-reorganization.

```
1. Num=length (Data)
2. Step=3
3. Cro=1
4.    For
          i=1; Num-Step-Cro+1
             res(i;:)=[reshape(Data(I;i+Step-1),1,Step),Data(i+Step+Cro-1)]
       End
5. Data_input=res(:,1:Step)
6. Data_output=res(:,Step+1,end)
```

Table 4
The specification of the 3.7 V/100 Ah lithium-ion battery.

Parameter	Value	Parameter	Value
Cell nominal capacity (Ah)	100	Peak discharge current	3C
Cell nominal capacity (V)	3.7	Maximum load current	2C
Charge cut-off voltage (V)	4.5 ± 0.05	Internal resistance (mΩ)	0.5–1
Discharge cut-off voltage (V)	2.75 ± 0.05	Working temperature (°C)	20–60
Standard charge current	1C	Dimension: l * w * h (mm)	148 * 27 * 93

Table 5
IRDPSO-FFBPNN parameter settings.

Experiment parameters	Setting
Dimension	30
Out-of-scope processing	Take the random values at the interval
Particle velocity dimension	[−100,100]
Particle position dimension	[−100,100]
Algorithm abort conditions	Reach the maximum value of iterations
Numbers of runs	51
Number of iterations	50
c_1	1.6
c_2	1.8

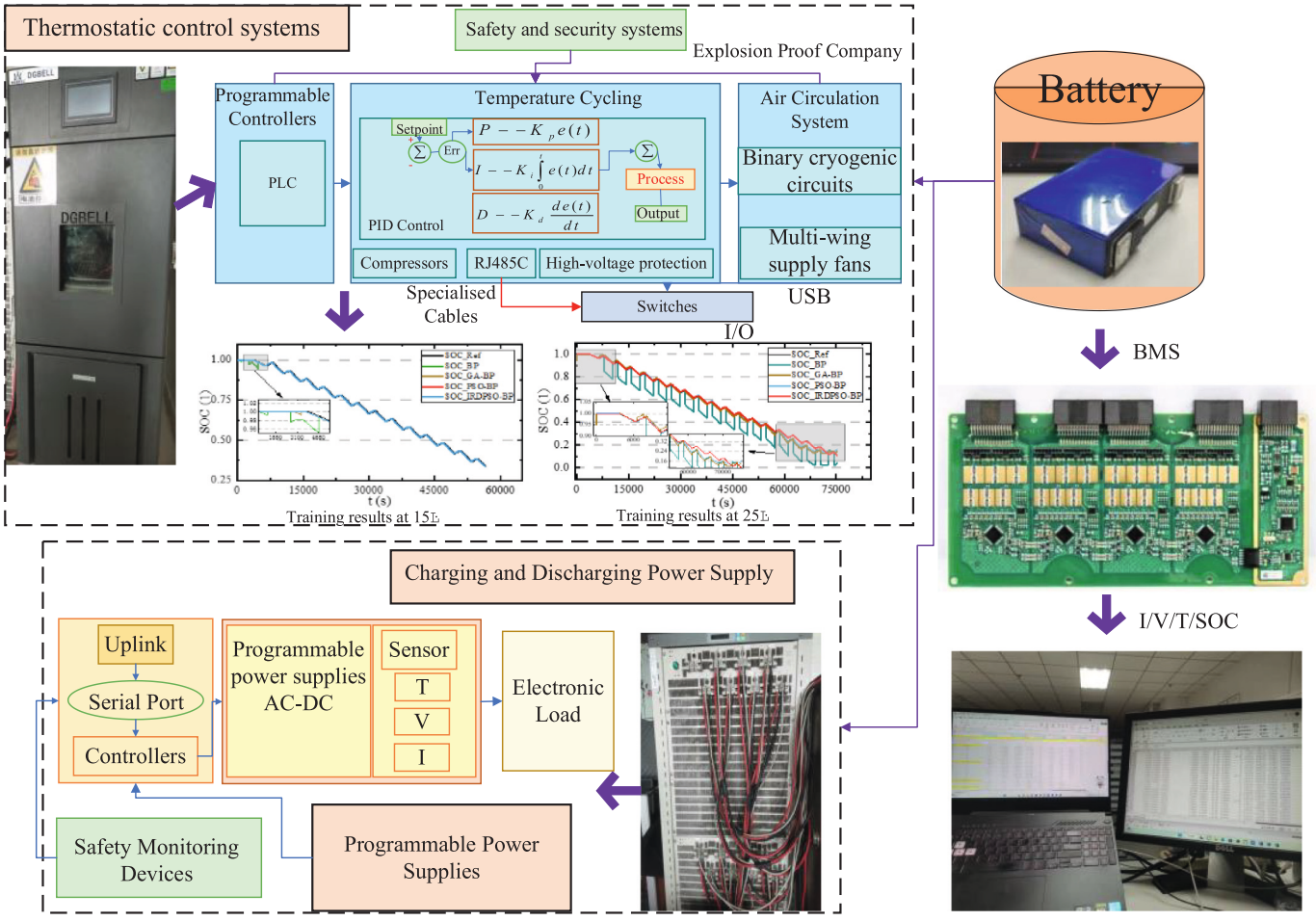


Fig. 6. The experimental platform establishment.

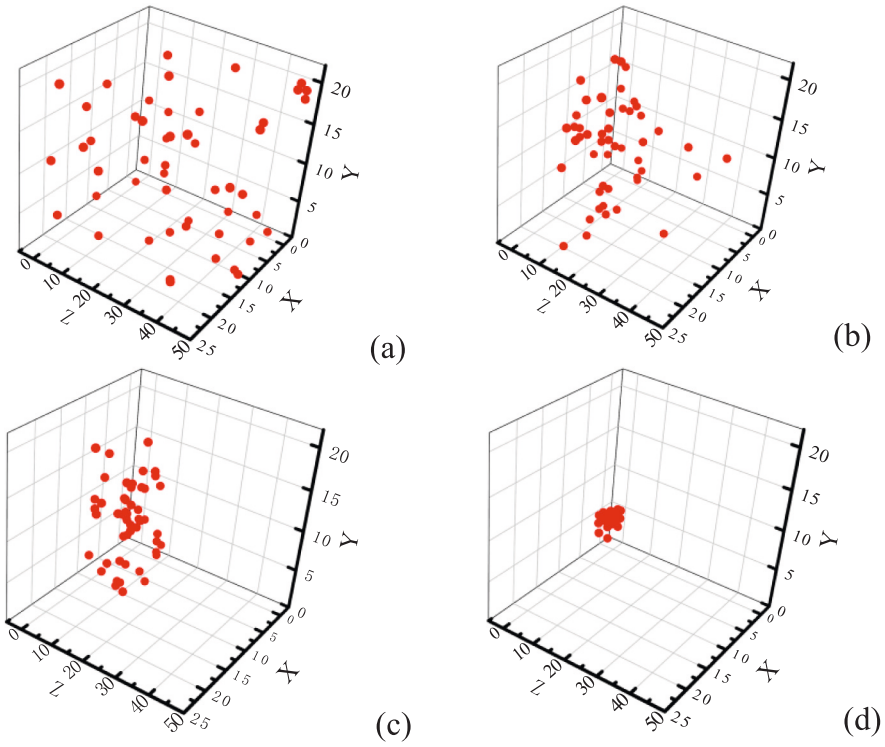


Fig. 7. The process of particles closes to the $P_{gbest.}$

As analyzed in the transformation from the traditional PSO algorithm to the RDPSO algorithm, the vocality is divided into random parts $VR^k_{i,n-1}$ and drift part $VD^k_{i,n-1}$, and the partial function of the initial weight w_P in the PSO algorithm is replaced by the random search component $VR^k_{i,n-1}$, which means the linear decrement of w_P is dis-integrated, and then another part function of w_P needs to continue to achieve the local search in the later iterations. The phenomenon of deviant trajectory is obvious from Eq. (20).

$$V_{i,n}{}^k = w_p V_{i,n-1}{}^k + c_1 r_{i,n-1}{}^k \left(P_{ibest,n-1}{}^k - X_{i,n-1}{}^k\right) + c_2 R_{i,n-1}{}^k \left(P_{gbest,n-1}{}^k - X_{i,n-1}{}^k\right) \tag{20}$$

The w_r represents the weight in the RDPSO algorithm, which needs to complete the search and guide the particle to close to the P_{gbest}, and the parameter β including the content of maximum and minimum value of w_r to limit the interval of the particle movement. The parameter σ is induced to mark the error in the process of closing to the P_{gbest}, and the value is recorded to guide the step and the direction to the next move of a certain particle. The expression with the σ of ultimate velocity is shown in Eq. (21).

$$\begin{cases} w_r(\sigma) = \beta_{i,n-1}{}^k r_{i,n-1}{}^k + w_{rmin} + \sigma^* r_{i,n-1}{}^k n_{i,n-1}{}^k \\ \beta_{i,n-1}{}^k = w_{rmax} - w_{rmin} \end{cases} \tag{21}$$

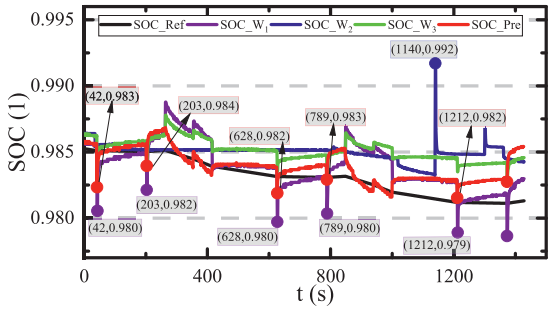
The $r^k_{i,n-1}$ is a random value that obeys the uniform distribution, the $n^k_{i,n-1}$ *is* the random value that obeys the normal distribution, the σ takes the value at the interval [0.2,0.5], which can describe the w deviation of a certain particle from its corresponding expectations. Parameter σ with directional components including x and y can guide the particle close to the P_{gbest} at an appropriate increment in its interval. And the illustration is shown in Fig. 5.

Accordingly, the expression of the $VR^k_{i,n-1}{}'$ can be updated as shown in Eq. (22).

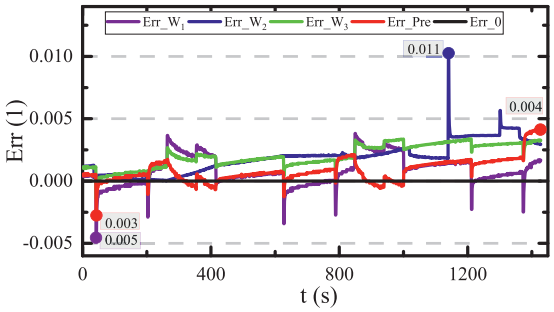
$$V_{i,n-1}{}^{k'} = VR_{i,n-1}{}^k + VD_{i,n-1}{}^k + w_r(\sigma) \tag{22}$$

(2) Cross-reorganization—achieve data regeneration and enhance the utilization

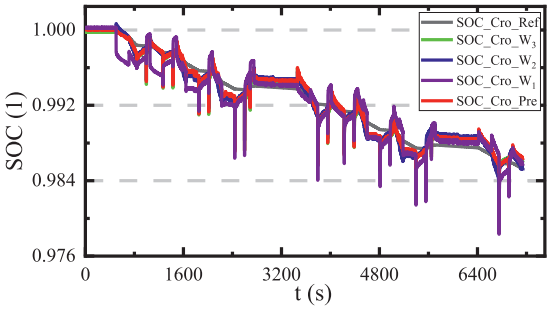
In contrast to the GA algorithm, RDPSO cannot generate new pop-ulations through genetic manipulation of selection, crossover, and mutation, which results in the limitation of the data utilization and the learning depth of the network. A method of data that updates the augment in each of three steps, and re-estimates each cross to traverse all test data, sets 75 %, and 25 % as training and testing data, respectively, and sets the cross-validate fold as 3, the max iteration is 50 to improve the data utilization. During the process, each data will complete two parts including being trained and being tested. In this way, the data volume will be three times greater than before, which improves the data utilization and learning depth, providing a mutual benefit to both the RDSPO algorithm and the FFBPNN. The code implementation of the cross-reorganization is shown in Table 3.



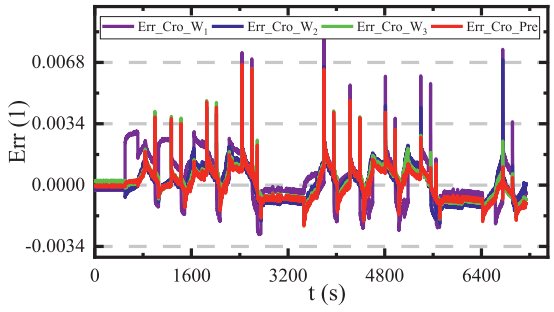
(a) The adjustment results of particles with σ



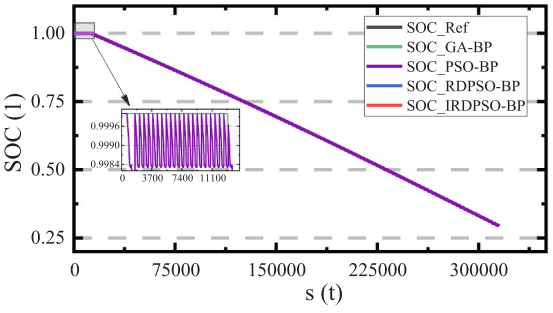
(b) The error curves with each σ



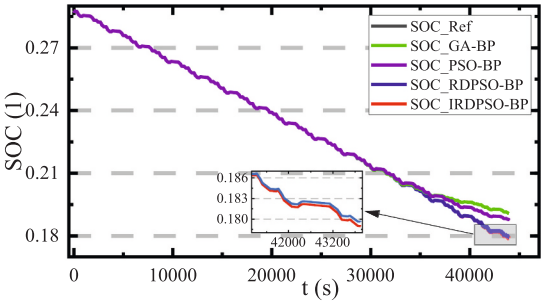
(c) The result curves of reorganization



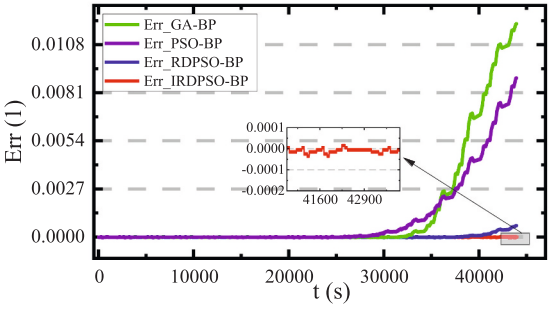
(d) The error curves of each reorganization data



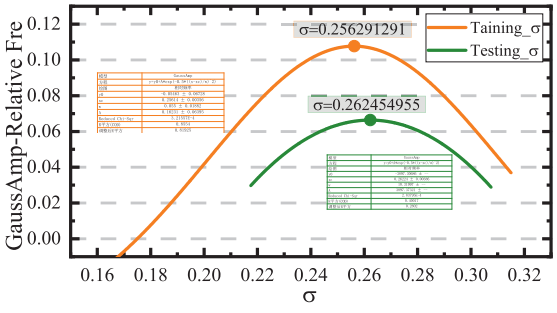
(e)Training results of neural networks



(f) Testing results of neural networks



(g)Error curves of testing results



(h) σ value of training and testing

Fig. 8. Verification results of weight error σ under BBDST working condition.

The step data is taken as the independent variable, and the cross value is passed on as the predicted data, which is based on the reading of the total amount of data.

3. Experimental testing and analysis

3.1. Test platform establishment

In the experiment, the 3.7 V/100 Ah ternary battery is set as the test object, and the Netware battery test equipment is the CT-4016-5V100A-NTFA, the constant temperature box is the DGBELL BTT—331C, the parameters of the battery is shown in Table 4 , and the parameter setting of the IRDPSO-FFBPNN is collected in Table 5. The experimental platform of the target lithium-ion battery test equipment is observed in Fig. 6.

3.2. Experimental analysis

The X-axis and the Y-axis represent the position of each particle, and the Z-axis means the iterations. The position of the P_{gbest} is set at (9,9,9) and particles are closing to the P_{gbest} . Specifically, it unveils a clustering phenomenon to make the search space get smaller, which can speed up the process of interacting with the particles and reduce the time needed to train. The convergence process in the RDPSO algorithm showed in Fig. 7.

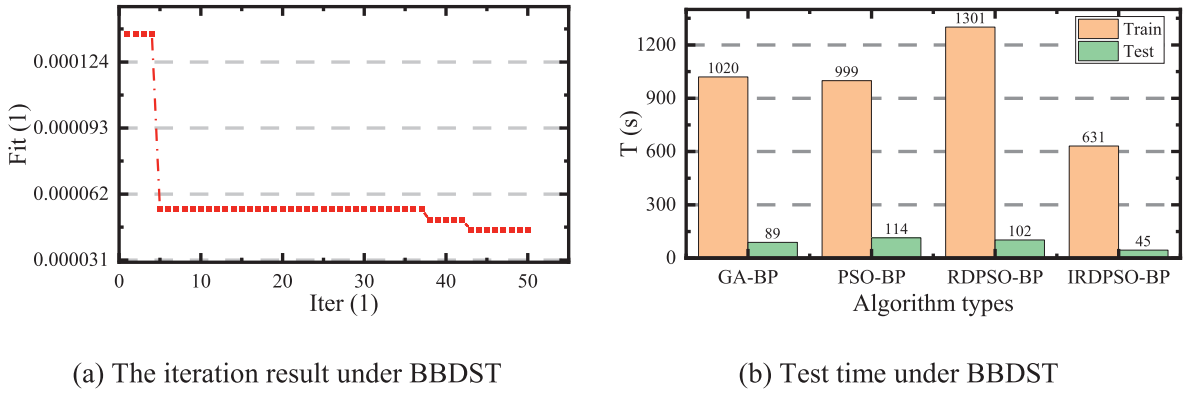
3.3. Verification analysis under BBDST working condition

The clustering phenomenon is converted to the test time reduction and the accuracy improvement in the data results, the efficiency of proposed methods are verified independently, and the results under BBDST working condition are shown in Fig. 8.

Fig. 8(a) and (b) presents the process of constant correction towards the SOC_Ref curve for different values of w(σ) from IRDPSO-FFBPNN, in which the peaks are decreasing, and the curve is smoothing. The ultimate adjustment result SOC Pre is almost overlapped with SOC Ref. In Fig. 8(c) and (d), the cross-reorganization was verified based on continuous correction of w(σ). In Fig. 8(e) and (f), training and test are compared with other algorithms based on FFBPNN, includes GA, PSO, and traditional RDPSO. The results show that all networks were able to maintain stability and high accuracy during the training process. In the testing phase, as the amount of data increased, all algorithms showed significant errors except IRDPSO, with the maximum errors reaching at 1.0879 %, 0.8179 %, 0.2167 % as Err_GA - BP, Err_PSO - BP Err_RDPSO - BP showed in Fig. 8(g) respectively, in comparison, the IRDPSO exhibits high stability and accuracy with a maximum error at 0.1021 %. In Fig. 8(h), the σ normal distribution value of training and test are recorded, they are 0.2563 and 0.2625, respectively, which represent the system results deviate little from their corresponding ex-pected values SOC_Ref. In addition to the above analysis, further information regarding the errors of the different algorithms and the time taken is given in Table 6.
In Table 6, the maximum error of the IRDPSO algorithm reached 0.1021 %, compared to GA 1.0879 %, PSO 0.8179 %, traditional RDPSO 0.2167 %, it achieved a high precision with the value at 0.9997 of R2. The iterations and time-consumption under DST are shown in Fig. 9.
In Fig. 9(a), the fitness F of the system satisfies the relation F = 1 Fit, the smaller the Fit value, the better the performance. Learning stops when the system reaches the conditional number of iterations, at which point the value of Fit is closed to 0.00003, which represents the IRDPSO-FFBPNN fitness reached 0.99997. The IRDPSO-FFBPNN performed a better in time-consumption as descripted in Fig. 9(b). With training data up to 300,000 in 631 s, the time is reduced by more than 36.8 % compared to other algorithms, and with test data over 40,000 in 45 s, the time is reduced by more than 49.4 % compared to other algorithms.

Table 6
The tests value of the neural networks under BBDST.

Indexes	Algorithm			
	GA-FFBP	PSO-FFBP	RDPSO-FFBP	IRDPSO-FFBP
Maximum error (%)	1.0879	0.8179	0.2167	0.1021
MAE (%)	2.2192	2.1265	2.2823	1.1319
RMSEP (%)	2.5367	2.2876	1.9730	1.3291
R^2	0.9114	0.9451	0.9759	0.9997



(a) The iteration result under BBDST (b) Test time under BBDST

Fig. 9. Iterations and time consumption under BBDST.

3.4. Verification analysis under DST working condition

To verify the IRDPSO-FFBPNN matching and stability under DST conditions, the training under 25 °C and test under 35 °C are completed and analyzed as shown in Fig. 10.

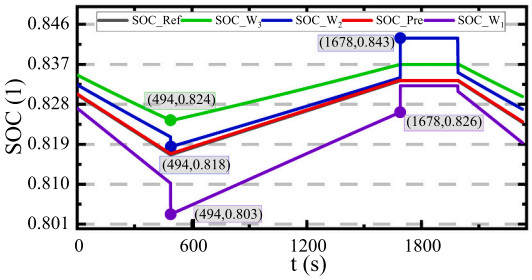
In Fig. 10(a) and (c), there are clear peaks and deviation from SOC_Ref of the first adjustment curve SOC_W_1 and SOC_Cro_W_1. With the action of the σ, the curves are continuously corrected towards until it reached to SOC_Pre and eventually becomes smooth, which can be observed about corresponding error curves in Fig. 10(b) and (d). The algorithms of BP, GA-BP, PSO-BP, and IRDPSO-BP are trained at 25 °C with high stability and accuracy in Fig. 10(e). Nevertheless, the results of test at 35 °C appeared obvious deviation from SOC_Ref, which maximum error reached 15 %, 6.6397 %, 3.1429 % of BP, PSO-BP, RDPSO-BP, respectively. The error curve of IRDPSO floats around 0 in Fig. 9(f) and (g). In Fig. 10(h), the σ normal distribution value of training and test are recorded as 0.2083 and 0.2280, represent a good repressiveness of system. Further information regarding the errors of the different algorithms and the time taken is given in Table 7.

From the above table, the optimization results of the FFBPNN using different algorithms show that the maximum error of the single BPNN is 15.0571 %, and the maximum error values of GA-BP, traditional RDPSO-BP are reduced to different degrees, by 3.1429 % and 3.1429 %, respectively. Nevertheless, the maximum error and R^2 of the IRDPSO- FFBPNN reached only 0.1237 % and 0.99996, reduced by 14.9334 %. The time-consumption under DST is shown in Fig. 11.

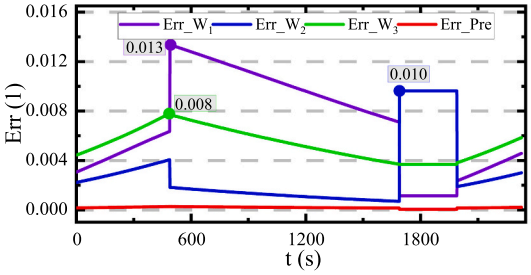
Learning stops at iteration 50 in Fig. 11(a), at which point the value of Fit is closed to 0.000063, and the fitness reached 0.99994 of system. In Fig. 11(b), with training data up to 60,000 in 92 s, the time is reduced by more than 57.9 % compared to other algorithms, and with test data over 75,000 in 116 s, the time is reduced by more than 33.3 % compared to other algorithms.

4. Conclusion

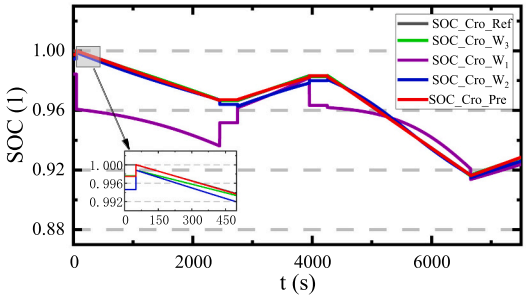
In this research, to achieve a high-efficient state of charge estimation of lithium-ion batteries, a novel three-layer feed forward backpropagation neural network is established, using the random drift particle swarm optimization algorithm to solve the low convergence and to avoid the local optima of the system, which uses voltage, current as the input, uses the value of the state of charge as the output. To solve the problems of particle crossover and insufficient data utilization of traditional random drift particle swarm optimization algorithm, the weight error and cross-reorganization are proposed and verified independently to form the improved random drift particle swarm optimization-backpropagation neural network which can enhance the effectiveness the information interaction for particles. Compared with other neural networks under complex working conditions, the improved random drift particle swarm optimization-backpropagation neural network performed the maximum error 0.1021 % in 45 s, 0.1237 % in 116 s under BBDST and DST, respectively, with high precision estimation in short time-consumption, which provides a theoretical reference for state of charge estimation of lithium-ion batteries.



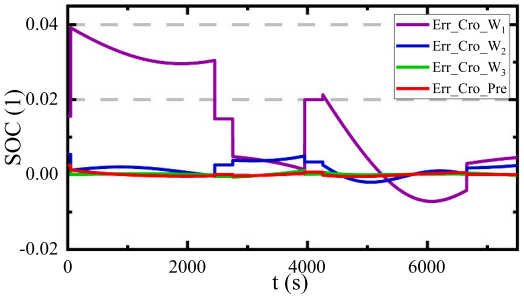
(a) The adjustment results of particles with σ



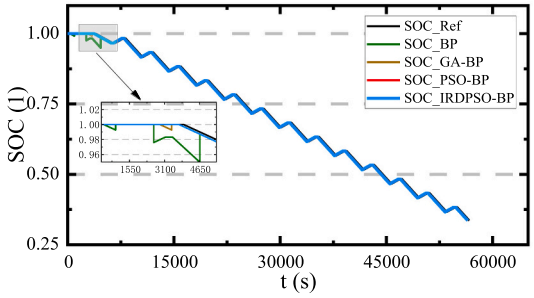
(b) The error curves with each σ



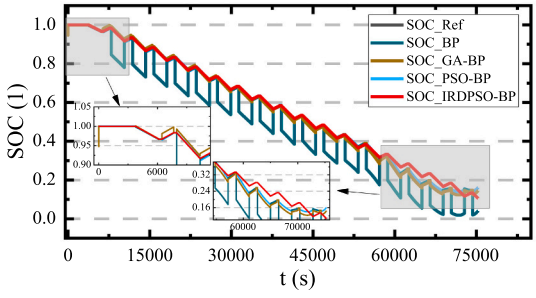
(c) The result curves of reorganization



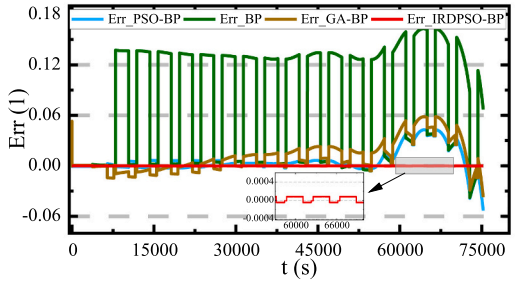
(d) The error curves of each reorganization data



(e)Training results at 25 °C of neural networks



(f) Testing results at 35°C of neural networks



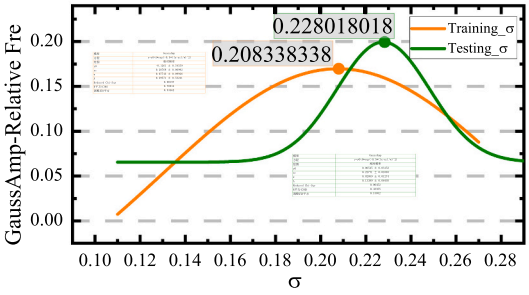
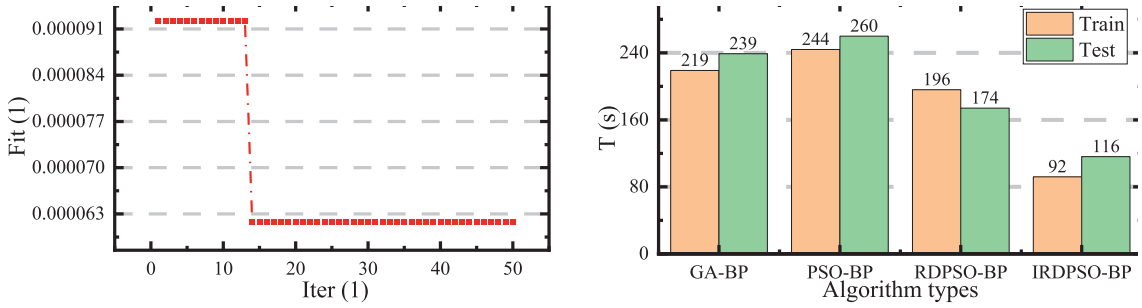


Fig. 10. Verification results of weight error σ under DST working condition.

Table 7
The tests value of the neural networks under DST.

Indexes	Algorithm			
	FFBP	GA-FFBP	RDPSO-FFBP	IRDPSO-FFBP
Maximum error (%)	15.0571	6.6397	3.1429	0.1237
MAE (%)	9.2192	6.1265	4.2823	1.1779
RMSEP (%)	11.5367	8.2876	5.0730	1.1368
R^2	0.53219	0.91878	0.924581	0.99996



(a) The iteration result under DST (b) Test time under DST

Fig. 11. Iterations and time consumption under DST.

CRediT authorship contribution statement

Nan Hai: Writing, Original draft preparation, Algorithm design, Data organization and analysis

Shunli Wang: Data curation, Conceptualization, Methodology **Donglei Liu**: Visualization, Investigation
Haiying Gao: Supervision, Investigation
Carlos Fernandez: Examine and verify, Software

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.

Data availability

Data will be made available on request.

References

[1] H. Shi, et al., Adaptive iterative working state prediction based on the double unscented transformation and dynamic functioning for unmanned aerial vehicle lithium-ion batteries, Meas. Control 53 (9–10) (2020).

[2] H. Shi, et al., Multi-time scale identification of key kinetic processes for lithium-ion batteries considering variable characteristic frequency, 2023, pp. 1211–1220.

[3] S. Adrian, et al., Impact of micro-cycles on the lifetime of lithium-ion batteries: an experimental study, J. Energy Storage 55 (PA) (2022).

[4] R. Chang, A review of the cathode materials development for lithium-ion batteries, 2022.

[5] X. Chunwu, et al., Comprehensive investigation on lithium batteries for electric and hybrid-electric unmanned aerial vehicle applications, Therm. Sci. Eng. Prog. 38 (2023) 3–7.

[6] L. Dezhi, et al., Electrochemical impedance spectroscopy based on the state of health estimation for lithium-ion batteries, Energies 15 (18) (2022).

[7] G. Dongliang, et al., State of health estimation for lithium-ion battery based on energy features, Energy 257 (2022) 12–16.

[8] L. Liu, et al., A highly scalable integrated voltage equalizer based on parallel- transformers for high-voltage energy storage systems, 2023, pp. 123–132.

[9] E., M.M, et al., A review of research needs in non-destructive evaluation for quality verification in electric vehicle lithium-ion battery cell manufacturing, J. Power Sources 561 (2023) 230–234.

[10] J, L.M, et al., Addendum: Elastomeric electrolytes for high-energy solid-state lithium batteries, Nature 609 (7927) (2022).

11] A.R. Sahu, et al., Online Approximation of SOC and temperature of a electric vehicle by combined OCV-CC method, 2021, pp. 265–269.

[12] Y. Jin, C. Su, S. Luo, Improved algorithm based on AEKF for state of charge estimation of lithium-ion battery 23, 2022, pp. 1003–1011.

[13] S., S.W, Emerging challenges for the safe transportation of lithium batteries, Process. Saf. Prog. 41 (3) (2022) 9–15.

[14] Q. Wenyu, C. Guici, Z. Tingting, An adaptive noise reduction approach for remaining useful life prediction of lithium-ion batteries, Energies 15 (19) (2022).

[15] J. Chen, et al., SOC estimation for lithium-ion battery using the LSTM-RNN with extended input and constrained output, 2022, pp. 909–912.

[16] D. Kavitha, N. Geetha, K. Duraisamy, Influence of electrode parameters on the performance behavior of lithium-ion battery, J. Electrochem. Energy Convers. Storage 20 (2022) 1–10.

[17] Artificial Neural Networks, Researchers' Work from McMaster University Focuses on Artificial Neural Networks (A comparison between Artificial Neural Network and Hybrid Intelligent Genetic Algorithm in predicting the severity of fixed object crashes among elderly drivers), in: Robotics & Machine Learning, 2020, pp. 342–345.

[18] Science - Operations Science, New operations science study findings have been reported from University of Porto (a mip model and a biased random-key genetic algorithm-based approach for a two-dimensional cutting problem with defects), J. Math. (2020) 222–234.

[19] Fuzzy Research, Researchers from Chang'an University Report New Studies and Findings in the Area of Fuzzy Research (Safety Evaluation System of Urban Traffic Network Based On Topological Genetic Algorithm), J. Robot. Mach. Learn. (2020) 1099–1112.

[20] W. Bing-Chuan, et al., A two-phase differential evolution for minimax optimization, Appl. Soft Comput. J. 131 (2022) 345–346.

[21] G.G. Velleda, et al., Investigation on the association of differential evolution and constructal design for geometric optimization of double Y-shaped cooling cavities inserted into walls with heat generation, J. Appl. Sci. 13 (3) (2023) 3–9.

[22] Z. Zhiqiang, et al., Improving differential evolution using a best discarded vector selection strategy, J. Inf. Sci. 609 (2022) 110–112.

[23] B. Soudan, M. Saad, IEEE, An evolutionary dynamic population size PSO implementation, in: 3rd International Conference on Information and Communication Technologies, 2008 (Damascus, Syria).

[24] D. Wang, H. Wang, L. Liu, Unknown environment exploration of multi-robot system with the FORDPSO, Swarm Evol. Comput. 26 (2016) 12–15.

[25] F. Habibi, et al., Utilizing RDPSO algorithm for economic-environmental load dispatch modeling considering distributed energy resources, Electrica 21 (3) (2021).

[26] J. Shao, S. Kumar, Optical backpropagation for fiber-optic communications using optical phase conjugation at the receiver, Opt. Lett. 37 (15) (2012) 3012–3014.

[27] Q. Wang, P.-J. Wu, J. Lian, SOC estimation algorithm of power lithium battery based on AFSA-BP neural network, 2020, pp. 121–125.

[28] L. Chao, S. Jun, P. Vasile, Diversity-guided Lamarckian random drift particle swarm optimization for flexible ligand docking, BMC Bioinf. 21 (1) (2020).

[29] Y. Wang, et al., Fractional order BPNN for estimating state of charge of lithium-ion battery under temperature influence 53, 2020, pp. 3707–3712.

[30] C. Wang, et al., A novel back propagation neural network-dual extended Kalman filter method for state-of-charge and state-of-health co-estimation of lithium-ion batteries based on limited memory least square algorithm, 2023, pp. 788–790.

[31] S. Jin, et al., A novel robust back propagation neural network dual extended Kalman filter model for state-of charge and state-of-health co-estimation of lithium-ion batteries, 2023, pp. 1–5.

[32] W. Xu, S. Wang, C. Jiang, et al., A novel adaptive dual extended Kalman filtering algorithm for the Li-ion battery state of charge and state of health co-estimation[J], Int. J. Energy Res. 45 (10) (2021) 14592–14602.

[33] G. Lian, et al., A BP neural network-Ant Lion Optimizer and UKF method for SOC estimation of lithium-ion batteries, J. Phys. Conf. Ser. 2369 (2022) 1211–1224.

[34] H. Shi, et al., On-line adaptive asynchronous parameter identification of lumped electrical characteristic model for vehicle lithium-ion battery considering multi- time scale effects, J. Power Sources 19 (2022) 12–16.

[35] H. Dai, C. MacBeth, Effects of learning parameters on learning procedure and performance of a BPNN, Neural Netw. (1997) 1505–1521.

[36] A.Y. Bhat, A.J.T.A. Qayoum, Viscosity of CuO nanofluids: experimental investigation and modelling with FFBP-ANN, Thermochim. Acta. 46 (19) (2022) 1200–1224.

[37] D. Jimenez Rezende, S. Mohamed, D. Wierstra, Stochastic backpropagation and approximate inference in deep generative models, in: International Conference on Machine Learning, 2014, pp. 267–278.

[38] J. Zhao, et al., Prediction of temperature and CO concentration fields based on BPNN in low-temperature coal oxidation 695, 2021, p. 178820.

[39] R. Dasgupta, Smooth saturation function-based position and attitude tracking of a quad-rotorcraft avoiding singularity, 2019, pp. 692–703.

[40] R. Vinoba, M.J.C.C. Vijayaraj, Novel control topology with obstacle detection using RDPSO-GBA in mobile AD-HOC network 160, 2020, pp. 847–857.

[41] Y. Li, et al., Fabric defect detection algorithm using RDPSO-based optimal Gabor filter 110, 2018, pp. 487–495.

[42] M. Dadgar, et al., RbRDPSO: Repulsion-based RDPSO for Robotic Target Searching 44, 2019, pp. 551–563.