CHEN, X., WANG, S., XIE, Y., FERNANDEZ, C. and FAN, Y. 2021. A novel fireworks factor and improved elite strategy based on back propagation neural networks for state-of-charge estimation of lithium-ion batteries. *International journal of electrochemical science* [online], 16(9), article 210948. Available from: https://doi.org/10.20964/2021.08.07

A novel fireworks factor and improved elite strategy based on back propagation neural networks for state-of-charge estimation of lithium-ion batteries.

CHEN, X., WANG, S., XIE, Y., FERNANDEZ, C. and FAN, Y.

2021

Int. J. Electrochem. Sci., 16 **(2021)** Article ID: 210948, doi: 10.20964/2021.08.07

International Journal of
ELECTROCHEMICAL
SCIENCE
www.electrochemsci.org

A novel Fireworks Factor and Improved Elite Strategy based on Back Propagation Neural Networks for state-of-charge estimation of lithium-ion batteries

Xianpei Chen[1], Shunli Wang[1,], Yanxing Xie[1], Carlos Fernandez[2], Yongcun Fan[1]*

[1] School of Information Engineering, Southwest University of Science and Technology, Mianyang 621010, China;
[2] School of Pharmacy and Life Sciences, Robert Gordon University, Aberdeen AB10-7GJ, UK.
[*]E-mail: wangshunli@swust.edu.cn

Received: 2 April 2021 / *Accepted:* 20 May 2021 / *Published:* 10 August 2021

The state of charge (SOC) of Lithium-ion battery is one of the key parameters of the battery management system. In the SOC estimation algorithm, the Back Propagation (BP) neural network algorithm is easy to converge to the local optimal solution, which leads to the problem of low accuracy based on the BP network. It is proposed that the Fireworks Elite Genetic Algorithm (FEG-BP) is used to optimize the BP neural network, which can not only solve the problem of the traditional neural network algorithm that is easy to fall into the local maximum optimal solution but also solve the limitation of the traditional neural network algorithm. The searchability of the improved algorithm has been significantly enhanced, and the error has become smaller and the propagation speed is faster. Combining the experimental data of charging and discharging, the proposed FEG-BP neural network is compared with the traditional genetic neural network algorithm (GA-BP), and the results are analyzed. The results show that the standard BP neural network genetic algorithm predicts error within 7%, while FEG-BP reduces the error to within 3%.

Keywords: Lithium-ion battery; State of charge; Genetic algorithm; Back Propagation; SOC estimation

1. INTRODUCTION

With the development of society, new energy is been sought to replace traditional fossil fuels, which has become the focus of attention of all countries. Lithium batteries have some advantages of high energy density, long life, high output power, and high-cost performance[1, 2]. The new energy has been widely used and developed in the field of new energy, greatly optimizing the energy system of today's society. On account of their green and environmental advantages, they have an important position in the field of renewable energy. With the increasing application of lithium batteries in the field of new energy, the ability of people to accurately estimate the SOC of lithium batteries is of great significance for giving full play to battery performance and realizing real-time status detection and safety control of lithium. battery. The SOC value cannot be measured directly[3-5]. Need to use physical quantities (such as voltage), current and indirect variables of temperature. Currently, commonly used algorithms include principle methods, modeling methods, and methods based on characteristic parameters.

The principle methods include the ampere integral method, internal resistance method, and open-circuit voltage method. These methods are easy to implement, do not need to construct a complex model, and artificially reduce the difficulty of the corresponding implementation. The influence of temperature and aging is great. The open-circuit voltage method is only suitable for SOC estimation in an offline state, and the hysteresis of the open-circuit voltage will increase the error of SOC estimation.

The method of establishing model estimation has high precision, and the commonly used methods are the Kalman filter method and synovial fluid observation method. Model-based algorithms improve the accuracy of lithium-ion battery SOC prediction to a certain extent, but the production process of battery models is complicated, and the accuracy of the model often increases the difficulty of corresponding calculations[6, 7]. Under the cycle life of the battery, the battery capacity or other physical quantities will change, especially the relationship between the open-circuit voltage and the SOC of the lithium-ion battery is transparent. The complexity increases accordingly[8].

The data-driven method does not require modeling, and the estimation accuracy is high. The typical representative of this kind of algorithm is the BP neural network algorithm, but only using a single BP neural network to estimate the SOC, it is difficult to meet the high-precision estimation requirements. One of the methods you can try is to optimize neural algorithms[9-11]. For example, you can reduce the error factors in the experiment process by strictly controlling external interference factors. The initial weight of the BP neural network and the prediction accuracy of the subsequent network have a greater impact. The optimization of the algorithm can improve tracking accuracy[12]. The complex electrochemical reactions inside the power lithium battery and the non-linear relationship between various factors that are constantly changing in operating conditions have led to large errors in traditional SOC estimation methods. The BP neural network algorithm is a new algorithm that simulates the learning skills of the human brain[13, 14]. No need to build an accurate mathematical model. Establish an output model by analyzing the corresponding relationship between input and output[15-17]. Since the BP neural network is a multi-layer feedback network based on backpropagation, the algorithm has a slower convergence speed and is easy to fall into local optimization. Genetic algorithm GA is an intelligent algorithm that can find the best solution by simulating natural selection and biological evolution mechanisms. It has strong convergence and robustness. It can be combined with neural networks to perfectly solve local optimal problems, and can greatly speed up network integration. Can greatly help us get the value we need[18, 19].

On this basis, a neural network algorithm based on the Fireworks Factor and a genetic algorithm with an improved elite strategy is proposed. First, introduce the fireworks coefficient[20-22]. The purpose of introducing the Fireworks Factor is to generate new individuals through the fireworks of the local optimal solution to make up for the lack of population diversity in the optimization process of the SGA (standard genetic algorithm, SGA) algorithm, thereby improving the analysis algorithm. site.

Global search function. For the elite strategy, consider the outstanding figures of this generation and the outstanding figures in the highly adaptable history as elites, and provide more crossover opportunities[23-27]. Based on the above two points, an improved algorithm is proposed, called Fireworks Elite Genetic Algorithm (FEG-BP)[28]. According to the experimental data of lithium-ion battery charging and discharging, different working conditions are used to estimate the SOC of the battery, and a variety of evaluation indicators are proposed to analyze different predictive performances.

2. MATHEMATICAL ANALYSIS

2.1 FEG-BP Algorithm

Elite talents are those with the highest adaptability during the development of the GA team. The genes in the genetic algorithm do not necessarily reflect the nature of the problem to be solved. Therefore, genes may not be independent of each other, and if they are only crossed, better combinations may be destroyed. In this way, the goal of good gene accumulation is not achieved, but the original good genes are destroyed. An elite retention strategy can prevent mixed operations from destroying the best individuals.

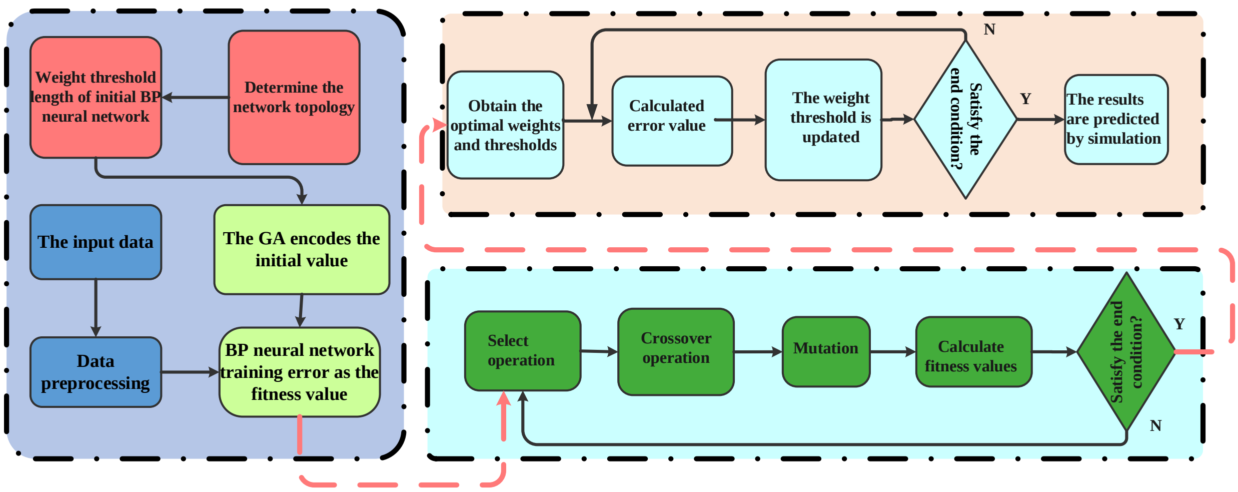


Figure 1. GA-BP algorithm flow chart

Population initialization and fitness function: The individual coding method is changed to real number coding, and each real number serves as a chromosome site, which can eliminate coding and decoding steps and simplify genetic operations. The real number string is divided into four parts, a single encoding method uses real number encoding, and the encoding content includes all weights and thresholds of the neural network.

Obtain the initial weights and thresholds of the BP neural network from the individual, and then use the training data to train the BP neural network to predict the output of the system, and then predict the absolute value of the error between the expected output and the target output. The predicted value is based on the individual fitness F, such as Eq. 1.

$$F = k(\sum_{i=1}^{n} abs(y_i - o_i)) \tag{1}$$

In the formula: where: n is the number of network output nodes; y_i is the expected output value of the *i*-th node of the BP neural network; o_i is the predicted output value of the *i*-th node; k is the corresponding coefficient. Selection operation and crossover operation: Since the individual uses real number coding when choosing the real number crossover method, the crossover operation of the Lth chromosome al and the *k*th chromosome kl at position *j* is as follows.

$$\begin{cases} p_i = \dfrac{f_i}{\sum_{j=1}^{n} f_i} \\ a_{kj} = a_{kj}(1-b) + a_{ij}b \\ a_{lj} = a_{lj}(1-b) + a_{kj}b \\ x_k = \dfrac{x_k - x_{\max}}{x_{\max} - x_{\min}} \end{cases} \tag{2}$$

Diversity avoids falling into the single-value trap. The fitness value of the *i*-th node of the neural network is *Fi*, and the number of individuals in the population is set to *N*. The roulette method can most intuitively and clearly express the randomness of the overall value of the function, which conforms to the characteristics of random inheritance of genetic algorithms.

Data normalization and mutation operation: To avoid the increase of the prediction error caused by the large difference in the order of magnitude between the sample data and the target data, the training sample data is first normalized, and the maximum-minimum method is used to normalize the training sample data. Among them, x_{min} and x_{max} are the minima and maximum values in the data sequence. Randomly select the *j*-th gene mutation of individual i, and the mutation results are shown in Eq.3: :

$$\begin{cases} a_{ij} = \begin{cases} a_{ij} + \left(a_{ij} - a_{\max}\right) * f(g) & r \geq 0.5 \\ a_{ij} + \left(a_{\min} - a_{ij}\right) * f(g) & r < 0.5 \end{cases} \\ f(g) = r_2 \left(1 - g / G_{\max}\right)^2 \end{cases} \tag{3}$$

In the formula, a_{min} is the lower limit of a_{ij}, a_{max} is the upper limit of gene a_{ij}, and r is a random number of [0,1]. a_{min} is the lower bound of the gene; g is the current number of generations; G_{max} is the maximum number of evolutions.FEG-BP model: The specific steps of the FEG-BP algorithm are shown in the figure:

Step1. Randomly initialize the population and the number of iterations t to generate the first generation population P_t with a size of *N*. Determine the number of iterations t, and observe whether the maximum number of iterations t_{max} is reached, if step 3 is not executed, otherwise step 8 is executed.

Step2. Calculate the fitness value in the solution set, use the roulette method to select, copy *N* excellent individuals, and form a population-only Q_t.Perform crossover and mutation operation to evolve the population (at this time, the population still becomes Q_t)

Step3. Select n (n≤10) local optimal solution scene fireworks operations (at this time, the population is still called Q_t). Combine P_t and Q_t to form a mixed population R_t, and implement the elite retention strategy. Generate a new population P_t with a scale of *N* from the mixed population R_t, at this time t=t+1, and execute step1.

Step4. Output the calculation result directly and end

2.2 Fireworks Factor algorithm principle

The cross operation of Fireworks Factors can easily lead to a jump in the solution set. Although the mutation operation can perform a local search, the mutation probability is small, resulting in fewer execution times of the mutation operation. Besides, since the mutation in the solution set is mostly harmful, the probability of mutation cannot be greatly increased. When solving non-convex functions, the global search ability of the algorithm is poor. For this reason, some people have proposed the FA operator to solve the problem of global composite function optimization. This operator represents a kind of explosive search process. From the perspective of the search algorithm, if you want to improve the search performance of the algorithm, you must perform an explosive search in the optimal solution part, and the number of individuals generated by the fireworks is large. On the contrary, the optimal solution is farther away from the result in fewer individuals to search. Since a single solution firework can produce multiple new individuals, too many solutions to be exploded will inevitably increase the amount of calculation and reduce the efficiency of algorithm optimization. Based on the above analysis, this paper selects n (n≤10) solutions from the solution set, performs an explosive search, and finally combines the previous generation solution set to form a "father-son mixed solution set" for selection, which effectively improves the algorithm's ability to explore new solution spaces ability. The calculation steps of solving the number of explosive individuals in xi are as follows:

Step1. According to the different pros and cons of individuals in the population, the production (the following one) generates corresponding subgroups.

$$C_i = \mu \bullet \frac{y_{\max} - f(x_i) + \varepsilon}{\sum_{i=1}^{n} \left(y_{\max} - f(x_i) \right) + \varepsilon} \tag{4}$$

Where: represents the number of individuals generated by the *i*-th solution; μ is a constant parameter that controls the total number of individuals generated by n solutions; y_{max} is the maximum value of the objective function in the solution set; ε represents the minimum constant, which is used to avoid division by zero errors.

Step2. Correct the Fireworks Factor.

To avoid too many or too few fireworks-generated individuals affect the efficiency and searchability of the algorithm, it needs to be modified. The correction function is defined as (Equation 1 below), and a and b are fixed parameters. In the above formula, a and b are fixed parameters. When $Ci>b\mu$, a<b<1 needs to be satisfied.

$$C = \begin{cases} \mathrm{round}(a \bullet \mu) & if \quad C_i < a\mu \\ \mathrm{round}(b \bullet \mu) & if \quad C_i > b\mu \\ \mathrm{round}(C_i) & otherwise \end{cases} \tag{5}$$

In the formula: C_i represents the displacement amplitude of the *i*-th solution fireworks; A_{max} represents the entire maximum fireworks amplitude; y_{min} is the minimum value of the objective function in the solution set; *h* represents the displacement distance of the fireworks; *rand(-1,1)* is one in the interval [-1,1] random number. The result is shown in Eq. (5).

$$\begin{cases} A = A_{max} \bullet \dfrac{f(x_i) - y_{min} + e}{\sum_{i=1}^{n} f(x_i) - y_{min} + e} \\ h = A_i \bullet rand(-1,1) \end{cases} \tag{6}$$

In the following formula, *x'(i, j)* represents a new entity created by the fireworks. If the new individual generated does not satisfy the solution domain space range. *x''(i, j)*is a new individual that meets the scope of the resolution domain after conversion. The result is shown in Eq. 6.

$$\begin{cases} x'(i,j) = h + x(i,j) \\ x''(i,j) = x_j^{\min} + \left| x'(i,j) \right| \% \left(x_j^{\max} - x_j^{\min} \right) \end{cases} \tag{7}$$

Algorithms based on improved genetic factors will eventually get better convergence, which can effectively prevent effective factors in genetic algorithms from losing excellent parts due to continuous genetic mutation.

BP neural network is composed of the input layer, hidden layer, and output layer. The first stage is the signal forward propagation; the input information is processed layer by layer through the input layer and the hidden layer and the actual output value of each unit is calculated; the second stage is the backpropagation of the error. If the expected value does not match it, the output will be output. The error calculation between the expected value and the expected value is used to adjust the weight parameters between each layer.

This paper chooses four influencing factors of lithium battery energy E, voltage U, current I, and resistance R as the input of the BP neural network, and the lithium battery SOC as the output. The return of errors is particularly important in the BP neural network. The network output error can be represented by the correlation function of the weight input layer weight V_{ij} and the hidden layer weight W_{jk}, as shown in Eq.8.

$$\begin{cases} \Delta W_{jk} = -\eta \, \partial E / \partial W_{jk} \\ \Delta V_{jk} = -\eta \, \partial E / \partial V_{ij} \end{cases} \tag{8}$$

The negative sign of the above formula in the direction of weight update, that is, the reverse of gradient descent. E is the square of the error between the expected output and the actual output. Eq.9 shows that the weight of the hidden layer is updated.

$$\begin{cases} W_{jk}(t+1) = W_{jk}(t) + \eta \delta_k y_j \\ W_{ij}(t+1) = W_{ij}(t) + \eta \delta_j x_i \\ m = \sqrt{n+l} + a \end{cases} \tag{9}$$

The η in the above formula is the learning efficiency, and the range is between (0,1). *δk, δj* are the error signals of the output layer and hidden layer, respectively. It reflects the continuous iterative process of network weight update. In this model, the number of nodes in the input layer is 4, and the number of nodes in the output layer is 1.

2.3 Improved Elite Strategy algorithm principle

Compared with traditional algorithms, genetic algorithms can not only work in coding mode, but also search multiple peaks in parallel, and do not perform operations on the parameters themselves. It has good operability, uses probabilistic transition rules instead of deterministic rules, and has a global optimization function.

This article uses the elite algorithm instead of the roulette algorithm. Generally, the elite algorithm has better performance. Among various genetic algorithms and genetic programs, the roulette method used by traditional genetic algorithms obtains the best solution through continuous crossover and genetics. The elite algorithm continues to increase the proportion of genetic elites to make the results better. The most important part of the elite strategy is elite retention and elite crossover.

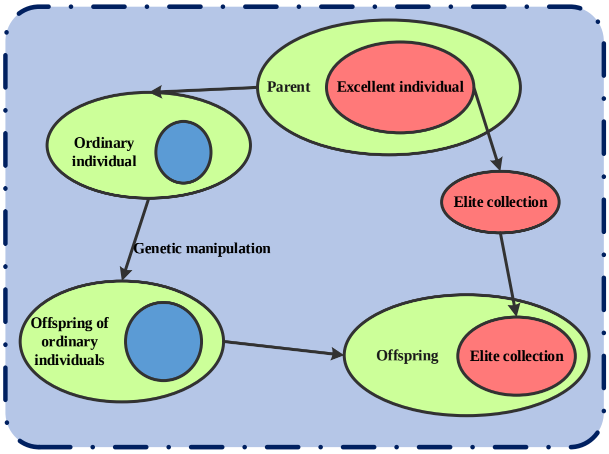


Figure 3. Elite retention strategy

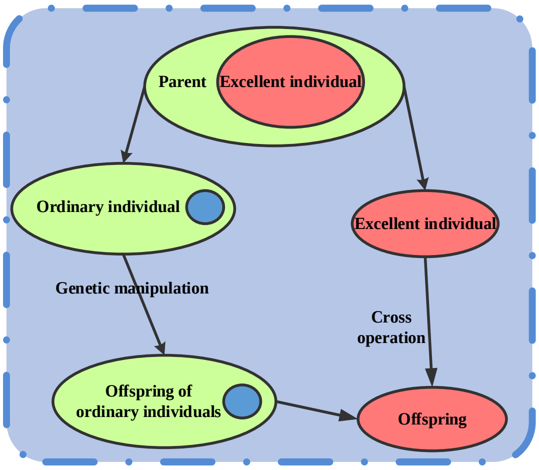


Figure 4. Elite crossover strategy

Excellent retention makes the outstanding individuals in this group not participate in crossover and mutation operations but directly left to the next generation, as shown in the figure. It can be found that it can effectively retain high-quality funds and prevent the destruction of longer good genes during crossover and mutation operations. However, the save operation is only for elites and has not been fully utilized. The flowchart of this algorithm is shown in Figure 3.

The elite divider retains the best solution of the previous generation of products. As shown in the figure, in addition to the traditional crossover operation for the group, the elite individuals and each individual in the group are also crossed according to the relevant probability. It provides more crossover opportunities for the elite group and increases the number of elite genes in the group. However, the elite individuals are not well preserved, causing them to be destroyed during the mutation process and cannot be used. The flowchart of this algorithm is shown in Figure 4.

The genetic algorithm is based on a large number of observations and summaries, and an algorithm that imitates the genetics and mutation of the biological world. The algorithm passed the observation of nature. By maintaining a set of possible solutions to simulate the biological evolution process of natural selection, the multi-directional search can be performed, and the formation and exchange of information in these directions can be supported. Compared with the search based on the point unit, the search based on the surface unit can better find the global optimal solution. The genetic search algorithm can simultaneously form multiple objects to reach the initial weight and threshold, thereby quickly optimizing the BP neural network so that the optimized BP neural network can better predict the function output. The elements of the genetic algorithm used to optimize the BP neural network include population initialization, fitness function, selection operation, crossover operation, and mutation operation.

2.4 Estimation strategy of lithium-ion battery SOC based on FEG-BP neural network

The specific calculation steps of the elite strategy in the t generation are as follows: (1) For the t generation group *St(X)*, set the function coverage to reach c_t; for each vector in $S_t(X)$, generate a random number, if the random number is less than c_t, then A vector in the elite set *Et(x)* crosses *St(X)*; if every vector of *Et(x)* can be used, the elements in *Et(x)* are recycled;

(2) For the *t*-th generation group St(x), after completing the crossover and mutation operations; calculate the fitness of each vector in [*St(X)*, *Et(x)*], and select all the vectors with the highest fitness to form an elite set *St+1(X)*.

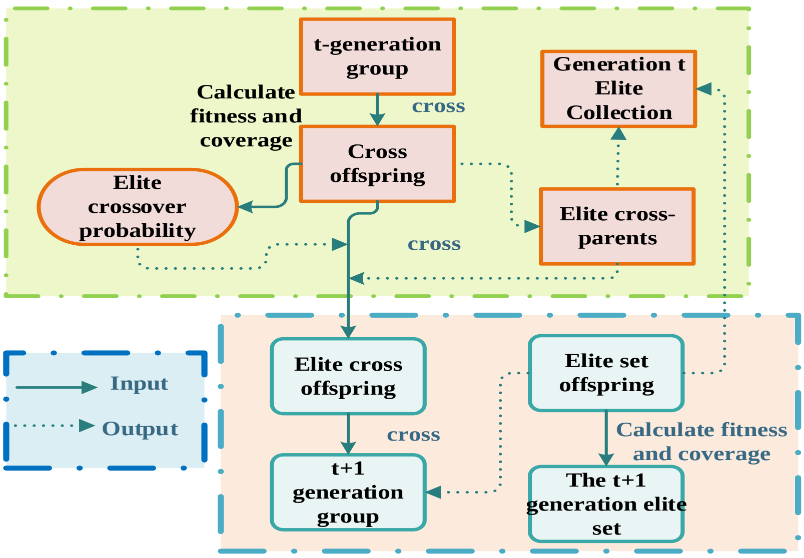


Figure 5. Improve the elite strategy flow chart

Comprehensive analysis of the advantages and disadvantages of the above two elite strategies, combined with the operation of functional coverage simulation in the actual process, proposes an elite strategy suitable for the field of functional verification, which can not only retain the excellent elite carrier but also make full use of it. The calculation process of the elite strategy in the t generation is shown in Figure 5.

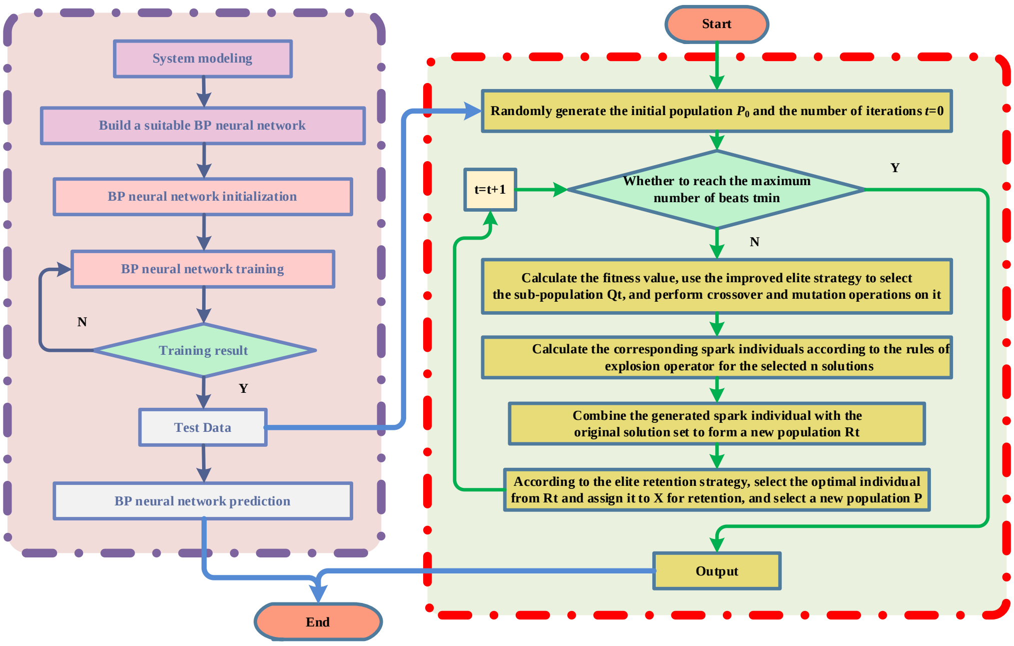


Figure 6. FEG-BP algorithm flow chart

The detailed process of the specific process is shown in Figure.6, where a variety of selection strategies can be used, among which the more well-known are the roulette method, competition strategy, elite strategy, and so on. Here, to further enhance the superiority selection function of the algorithm, the elite strategy is chosen. But the first step, as the initial stage, requires raw data values and can only use the roulette method.

3. EXPERIMENTAL RESULT

3.1 Platform construction

The experiment uses a ternary polymer macrocell power lithium battery with a charging cut-off voltage of 4.2V, a capacity of 70Ah, 4.5V as the protection voltage, and verification at 25°C. The following verification data are collected on the platform shown in figure 7.

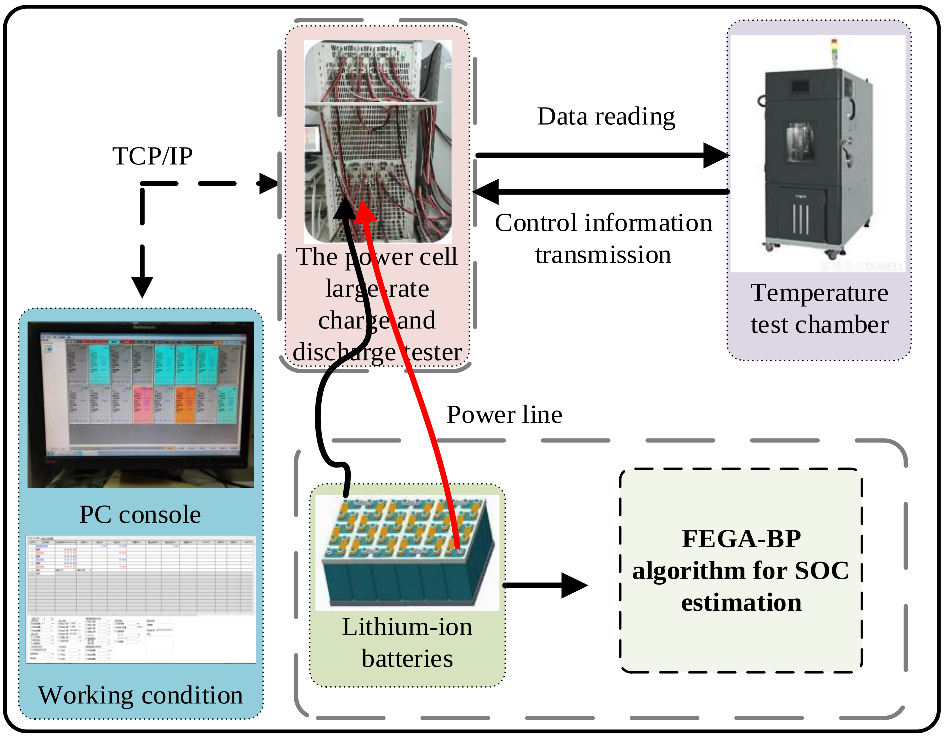


Figure 7. Test platform construction

After connecting the battery and the battery test system (NEWARE BTS-4000), debug the corresponding basic data through the upper computer position to set different charging and discharging conditions and measure the corresponding current, voltage, energy, and capacity of the battery under unused conditions, The flow chart is shown in Figure 5. The corresponding data is measured to facilitate further corresponding verification of the algorithm.

After setting up the corresponding detection platform, use different algorithms for training, and finally draw more relevant conclusions. First measure the initial capacity of the battery to get the initial SOC value, this time the initial value of SOC is 100%. Part of the conclusion data is shown in Tab1. In the following precise and complex working conditions, the comparison between FEG-BP and other algorithms is carried out. According to the display on the graph, the excellent robustness of this algorithm can be triggered[26, 29].

Table 1. Part of the sample data

Step	*T*/°C	*I*/A	*U*/V	*SOC*
1	25.0	-70	4.1961	1
2	25.0	-70	4.0060	0.9
3	25.0	-70	3.8789	0.8
4	25.1	-70	3.7636	0.7
5	25.0	-70	3.6675	0.6

3.2 DST working condition verification and result analysis

In practical applications, the real-time current of lithium-ion batteries is complex and variable. Under different working conditions, the current often suddenly switches and stops, which puts forward strict requirements on the dynamic performance of the battery, and also brings difficulties to the SOC estimation of the lithium-ion battery under complex working conditions. To further verify the SOC estimation model of lithium-ion batteries under more complex application conditions, the model has been simulated and verified through custom DST experimental data. The current and voltage of DST are shown in Figure 8.

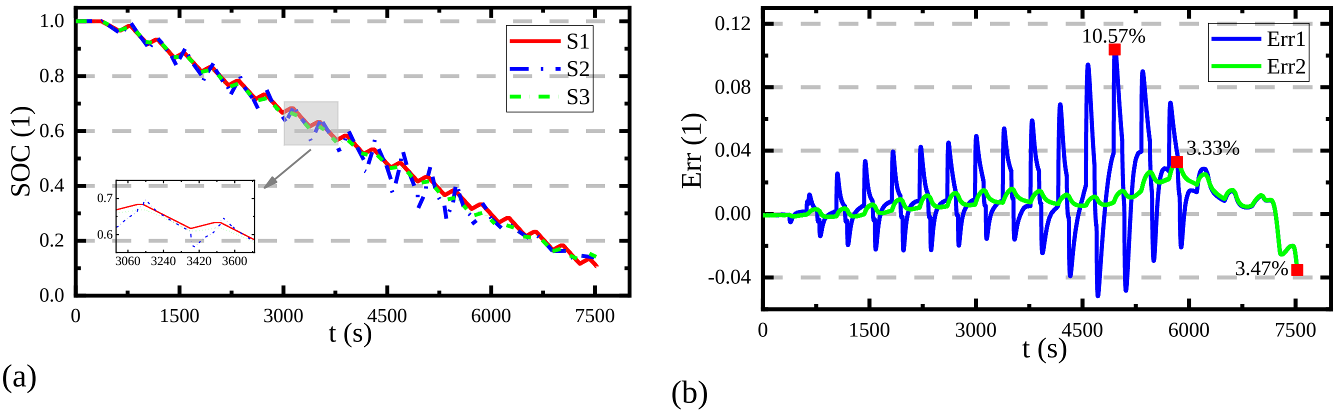
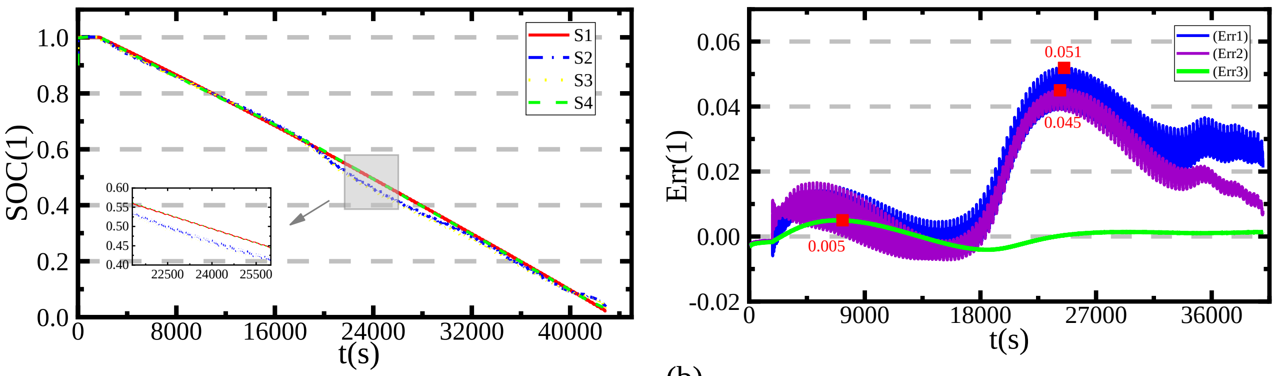


Figure 8. SOC verification under DST conditions

In Figure 8(a), S1 is the SOC value calculated by the ampere integral method, S2 is the SOC value calculated by the EKF algorithm, and S3 is the SOC value calculated by the FEG-BP algorithm. In Figure 8(b), Err1 is the EKF error, and Err3 is the FEG-BP error. Among them, the maximum error of SOC of FEG-BP is 3.47%, the maximum error of SOC of EKF is 10.57%. FEG-BP algorithm has a better effect in estimating SOC.

3.3 BBDST working condition verification and result analysis

For the BP neural network, this experiment uses the Beijing Public Transport Dynamic Stress Test (BBDST) for model training. The BBDST condition is very complicated and can reflect and better training data. Using offline EKF, the current and voltage under BBDST conditions at this time are used as inputs, and the first-order model parameters corresponding to the SOC value at this time are added as inputs to form a five-input model. The SOC obtained by the ampere-hour integration method is used as an accurate output. The network model trained by BP neural network can be obtained.



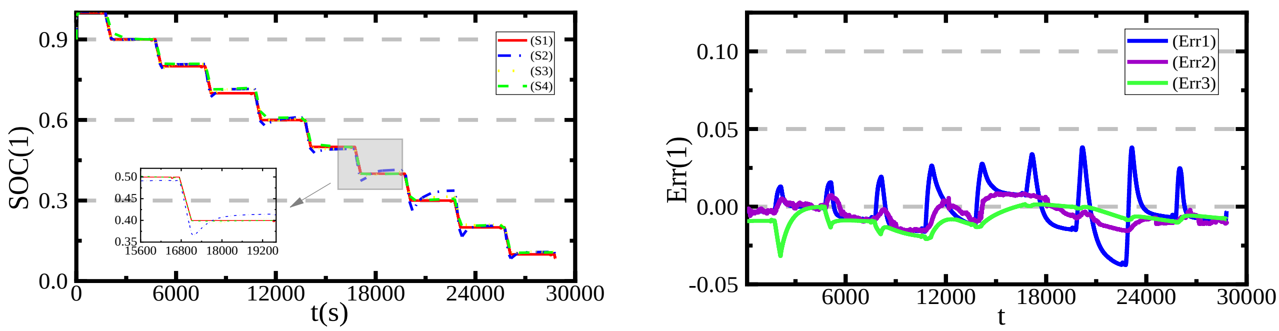
(a) (b)

Figure 9. SOC verification under BBDST conditions

In Figure 9(a), S1 is the SOC value calculated by the ampere integral method, S2 is the SOC value calculated by the EKF algorithm, S3 is the value calculated by AEKF, and S4 is the SOC value calculated by the FEG-BP algorithm. In Figure 9(b), Err1 is the EKF error, Err2 is the size of the AEKF error, and Err3 is the FEG-BP error. Among them, the maximum error of SOC of FEG-BP is 0.5%, the maximum error of SOC of EKF is 5.1%, and the maximum error of SOC of AEKF is 4.5%.

3.4 HPPC working condition verification and result analysis

To verify the accuracy of the improved FEG-BP algorithm for lithium-ion battery SOC estimation, capacity test experiments and HPPC experiments are used to verify the estimation. Use the trained network model and the first-order electronic circuit model. At the same time, the experiment takes the ampere-hour integration method and the extended Kalman filter algorithm as a reference, as shown in Figure 10.



(a) (b)

Figure 10. SOC verification under HPPC conditions

In Figure.9(a), SOC1 is the SOC value calculated by the ampere integral method, SOC2 is the SOC value calculated by the GA-BP algorithm, and SOC3 is the SOC value calculated by the FEG-BP algorithm. In Figure.9(b), Err1 is the GA-BP error, and Err2 is the FEG-BP error. Before disregarding the divergence of the final result, the SOC maximum error of FEG-BP is 1.28%, and the maximum error of GA-BP SOC is 3.20%.

4. CONCLUSIONS

In this paper, the Fireworks Factor is added to the original basis of the genetic algorithm, and the Fireworks Factor can be well adapted to the neural network algorithm. The Fireworks Factor can generate a local population in the original solution set, which increases the use range of the algorithm. At the same time, it further optimizes the calculation space of the algorithm, forming a genetic algorithm of “multi-population”, which can better avoid the premature algorithm of the algorithm and make the population in In the process of evolution, it has stronger competitiveness and global optimal solution ability, effectively avoiding falling into the local optimal solution situation. At the same time, the introduction of the elite retention strategy, which is one of the most representative strategies in the genetic algorithm, uses extreme methods to ensure that the effective factors can enter the next iteration to prevent methods such as the roulette strategy from causing the optimal solution to be abandoned, thereby affecting the overall experimental effect. By comparison, the improved FEG-BP and the KKF, AEKF, the ampere-hour integration method is used for SOC processing under multiple working conditions, and the final SOC difference is compared, it can be easily concluded that the algorithm proposed in this article has better. The global search capability, search performance, and calculation accuracy. However, the author found in the course of the experiment that the algorithm requires a large amount of data training in the early stage and takes a long time. It needs further research and improvement to improve its computing power.

ACKNOWLEDGMENTS
The work was supported by the National Natural Science Foundation of China (No. 61801407), Sichuan science and technology program (No. 2019YFG0427), China Scholarship Council (No. 201908515099), and Fund of Robot Technology Used for Special Environment Key Laboratory of Sichuan Province (No. 18kftk03).

References

1. T. Hirasawa, M. Yoshida and S. y. Obara, *International Journal of Energy Research*, 45 (2021) 807.
2. W. Zhao, X. Kong and C. Wang, *Proceedings of the Institution of Mechanical Engineers Part D-Journal of Automobile Engineering*, 232 (2018) 357.
3. M. Brenna, F. Foiadelli, C. Leone and M. Longo, *Journal of Electrical Engineering & Technology*, 15 (2020) 2539.
4. B. Wang, F. Qin, X. Zhao, X. Ni and D. Xuan, *International Journal of Energy Research*, 44 (2020) 4812.
5. M. S. H. Lipu, M. A. Hannan, A. Hussain, M. H. Saad, A. Ayob and M. N. Uddin, *Ieee Transactions on Industry Applications*, 55 (2019) 4225.

6. L. Cai, J. Meng, D.-I. Stroe, J. Peng, G. Luo and R. Teodorescu, *Ieee Transactions on Power Electronics*, 35 (2020) 11855.
7. M. F. B. Binelo, A. T. Z. R. Sausen, P. S. Sausen, M. O. Binelo and M. de Campos, *International Journal of Energy Research*, 45 (2021) 1023.
8. M. A. Hannan, M. S. H. Lipu, A. Hussain, M. H. Saad and A. Ayob, *Ieee Access*, 6 (2018) 10069.
9. H. Dai, G. Zhao, M. Lin, J. Wu and G. Zheng, *Ieee Transactions on Industrial Electronics*, 66 (2019) 7706.
10. H. Yao, X. Jia, Q. Zhao, Z.-J. Cheng and B. Guo, *Ieee Access*, 8 (2020) 95333.
11. A. Karthik, P. Kalita, A. Garg, L. Gao, S. Chen and X. Peng, *International Journal of Green Energy*, 18 (2021) 80.
12. M. S. Patil, J.-H. Seo, S. Panchal and M.-Y. Lee, *International Journal of Energy Research*, 45 (2021) 2533.
13. R. Xiong, R. Yang, Z. Chen, W. Shen and F. Sun, *Ieee Transactions on Industrial Electronics*, 67 (2020) 1081.
14. F. Liu, X. Liu, W. Su, H. Lin, H. Chen and M. He, *International Journal of Energy Research*, 44 (2020) 6338.
15. W. Xiong, Y. Mo and C. Yan, *Ieee Access*, 9 (2021) 1870.
16. Y. Liu, X. Wu, J. Du, Z. Song and G. Wu, *Renewable Energy*, 147 (2020) 2470.
17. D.-C. Lee, K.-J. Lee and C.-W. Kim, *International Journal of Precision Engineering and Manufacturing-Green Technology*, 7 (2020) 829.
18. Z. Xue, Y. Zhang, C. Cheng and G. Ma, *Neurocomputing*, 376 (2020) 95.
19. L. Chen, Y. Zhang, Y. Zheng, X. Li and X. Zheng, *Neurocomputing*, 414 (2020) 245.
20. T. Sun, B. Xu, Y. Cui, X. Feng, X. Han and Y. Zheng, *Journal of Power Sources*, 484 (2021) 229248.
21. T. Zahid, K. Xu, W. Li, C. Li and H. Li, *Energy*, 162 (2018) 871.
22. M. S. H. Lipu, M. A. Hannan, A. Hussain, M. H. M. Saad, A. Ayob and F. Blaabjerg, *Ieee Access*, 6 (2018) 28150.
23. L. Li, M. Hu, Y. Xu, C. Fu, G. Jin and Z. Li, *Applied Sciences-Basel*, 10 (2020) 6371
24. J. Shen, J. Xiong, X. Shu, G. Li, Y. Zhang, Z. Chen and Y. Liu, *International Journal of Energy Research*, 45 (2021) 5586.
25. Q. Zhu, M. Xu, W. Liu and M. Zheng, *Energy*, 187 (2019) 115880
26. M. Wei, M. Ye, J. B. Li, Q. Wang and X. Xu, *Ieee Access*, 8 (2020) 189236.
27. Q. Yang, J. Xu, X. Li, D. Xu and B. Cao, *International Journal of Electrical Power & Energy Systems*, 119 (2020) 105883
28. C. Liu, Y. Gao and L. Liu, *International Journal of Energy Research*, 45 (2021) 2303.
29. R. Wang, Y. Sun, V. Gvozdetskyi, X. Zhao, F. Zhang, L.-H. Xu, J. V. Zaikina, Z. Lin, C.-Z. Wang and K.-M. Ho, *Journal of Applied Physics*, 127 (2020) 904902