



OPEN

# Faults locating of power distribution systems based on successive PSO-GA algorithm

Wenzhang Xu<sup>1</sup>, Jiachun Li<sup>1</sup>✉, Lv Yang<sup>1</sup>✉ & Quan Yu<sup>2</sup>

As the terminal of the power system, the distribution network is the main area where failures occur. In addition, with the integration of distributed generation, the traditional distribution network becomes more complex, rendering the conventional fault location algorithms based on a single power supply obsolete. Therefore, it is necessary to seek a new algorithm to locate the fault of the distributed power distribution network. In existing fault localization algorithms for distribution networks, since there are only two states of line faults, which can usually be represented by 0 and 1, most algorithms use discrete algorithms with this characteristic for iterative optimization. Therefore, this paper combines the advantages of the particle swarm algorithm and genetic algorithm and uses continuous real numbers for iteration to construct a successive particle swarm genetic algorithm (SPSO-GA) different from previous algorithms. The accuracy, speed, and fault tolerance of SPSO-GA, discrete particle swarm Genetic algorithm, and artificial fish swarm algorithm are compared in an IEEE33-node distribution network with the distributed power supply. The simulation results show that the SPSO-GA algorithm has high optimization accuracy and stability for single, double, or triple faults. Furthermore, SPSO-GA has a rapid convergence velocity, requires fewer particles, and can locate the fault segment accurately for the distribution network containing distorted information.

**Keywords** Power distribution network, Fault location, Particle swarm algorithm, Genetic algorithm, Hybrid algorithm

As the terminal of the power system, the distribution network is characterized by many branches, variable operation modes, a large number of devices, and poor operation conditions, which makes it the main area where distribution system failures occur. With the access of distributed generation, the traditional distribution network has changed from a single-power radiant structure to a multi-power structure, and the traditional fault location algorithm based on single-power supply is no longer applicable, so further exploring new fault location methods is necessary. In addition, due to the installation characteristics of the feeder terminal unit (FTU), the information uploaded by FTU may be distorted. Therefore, the fault location algorithm should also meet the requirements of positioning accuracy and high fault tolerance<sup>1</sup>.

At present, the fault location methods used in distributed power distribution networks include intelligent algorithms<sup>2,3</sup>, matrix location method<sup>4</sup>, and synchronous vector method<sup>5</sup>. The matrix location algorithm usually needs to establish a structure matrix related to the current distribution network and then carry out specific operations with the fault information matrix to obtain the fault information of the distribution network. Literature<sup>6</sup> proposes a fault location model combining matrix and particle swarm optimization algorithms. Although the positioning accuracy and anti-interference ability are improved compared with the traditional matrix algorithm, it can also cause a dimensional disaster. The synchronous vector method can realize high-precision fault location by providing real-time current and voltage, but the synchronous vector unit (PMU) equipment is expensive<sup>7</sup>. Therefore, some authors<sup>8</sup> have effectively mitigated this defect by considering the impact of load loss on PMU equipment layout.

In recent years, the rapid development of artificial intelligence technology has provided a reference for the solution of fault localization problems in distribution networks with multiple power structures. Among them, well-studied intelligent algorithms include particle swarm optimization (PSO)<sup>9</sup>, genetic algorithm (GA)<sup>10</sup>, artificial fish swarm algorithm (AF)<sup>11</sup>, and so on. However, a single PSO or GA algorithm often tends to fall into the dilemma of local optimal solution<sup>12</sup>. For this reason, some researchers<sup>13</sup> proposed a switching function expression

<sup>1</sup>School of Mechanical Engineering, Guizhou University, Guiyang 550000, Guizhou, China. <sup>2</sup>Zunyi Long March Power Distribution Equipment Co. LTD, Zunyi 563000, Guizhou, China. ✉email: jcli@gzu.edu.cn; lyang6@gzu.edu.cn

for multiple faults of multiple power sources and utilized a hybrid discrete PSO-GA algorithm better to solve the fault localization problem in distribution networks. However, the algorithm needs to improve its iteration speed triggered by the unsuccessful speed update, and it is prone to falling into the local optimum. Although the binary artificial fish swarm algorithm has an excellent performance in localization effect, the algorithm is complex, and the optimality-seeking speed is slow. In contrast, the continuous particle swarm algorithm can overcome the problem of speed update failure in binary iterative optimization search and has a high iteration speed. Therefore, successive particle swarm genetic algorithm (SPSO-GA) is established in this paper and takes the IEEE 33-node distributed distribution network as an example to compare and study the performances of SPSO-GA, PSO-GA, and AF in terms of localization accuracy, arithmetic speed, and fault tolerance.

## Distribution network fault location rules

### State encoding style

There are only fault and normal states in the feeder segment of the distribution network. Therefore, a 1 or 0 code can represent a specific segment's feeder state. The expression is as follows:

$$X_i = \begin{cases} 1, & \text{segment } i \text{ is faulty} \\ 0, & \text{segment } i \text{ is not faulty} \end{cases} \quad (1)$$

After the DG is connected, the feeder current direction changes to bidirectional. When a fault occurs, the FTU at the node can detect the fault current, and the node status code can be expressed as:

$$I_j = \begin{cases} 1, & \text{Node } j \text{ detects fault current in the positive direction} \\ -1, & \text{Node } j \text{ detects fault current in the negative direction} \\ 0, & \text{No fault current is detected at node } j \end{cases} \quad (2)$$

### Switching function

The switch function expresses the connection between switch information and each feeder segment. The switch function in the distribution network containing DG is:

$$I_j^*(x) = I_{ju}(x) - I_{jd}(x) \quad (3)$$

where

$$I_{ju}(x) = \left\{ \prod_u^{M_1} [K_u (1 - \prod_{j,s_u} X_{j,s_u})] \right\} \times \prod_{j,d}^N X_{j,d} \quad (4)$$

$$I_{jd}(x) = \left\{ \prod_d^{N_1} [K_d (1 - \prod_{j,s_d} X_{j,s_d})] \right\} \times \prod_{j,u}^M X_{j,u} \quad (5)$$

where switch  $j$  is the breaking point, the distribution network is divided into upstream and downstream, i.e.,  $u$  and  $d$  two parts, the upstream and downstream lines have  $M_1$  and  $N_1$  power supplies respectively; “ $\prod$ ” means “logical or”;  $I_j^*(x)$  is the switch function of the  $j$ th switch,  $I_{ju}(x)$  and  $I_{jd}(x)$  are the switching functions of the upstream and downstream lines of the  $j$ th switch;  $K_u$  and  $K_d$  are the power switching coefficients of the upstream and downstream lines, which are taken as 1 if the power supply is connected and 0 otherwise;  $X_{j,s_u}$  and  $X_{j,s_d}$  are the status values of the feeder section from the  $j$ th switch to the upstream power supply  $S_u$  and the downstream power supply  $S_d$ ;  $X_{j,u}$  and  $X_{j,d}$  are the states of the upstream and downstream feeder sections, respectively;  $M$  and  $N$  are the total number of feeder sections of the upstream and downstream lines, respectively.

### Fitness function

The fitness function is a function used to compare ideal state value and actual value, and its expression is as follows<sup>14</sup>:

$$f(x_j) = \sum_{j=1}^n |I_j - I_j^*| + \omega \sum_{j=1}^n X_j \quad (6)$$

where  $I_j$  is the actual reported information of FTU at switch  $j$ , and  $I_j^*$  is the expected status value of FTU at switch  $j$ ;  $n$  is the total number of extents and nodes;  $X_j$  represents the state of the  $j$  section;  $\omega$  is the anti-misjudgment weight coefficient. The second term on the right can solve the case of multiple solutions of the function. The smaller the fitness function value, the closer the predicted fault segment is to the actual fault segment.

## Distribution network fault location algorithm

### Improvement of particle swarm optimization algorithm

The particle swarm optimization model is built to imitate the foraging process of birds<sup>15</sup>, and its binary particle speed and position update expression is as follows:

$$v_{id}(t+1) = \omega v_{id}(t) + c_1 r_1 (pbest_{id}(t) - x_{id}(t)) + c_2 r_2 (gbest_{id}(t) - x_{id}(t)) \quad (7)$$

$$x_{id}(t+1) = \begin{cases} 1, & \text{rand} < \text{sign}(v_{id}(t+1)) \\ 0, & \text{other} \end{cases} \quad (8)$$

$$\text{sign}(v_{id}(t+1)) = \begin{cases} 0.98, & 4 < v_{id}(t+1) \\ \frac{1}{1+e^{-v_{id}(t+1)}}, & -4 \leq v_{id}(t+1) \leq 4 \\ -0.98, & v_{id}(t+1) < -4 \end{cases} \quad (9)$$

In the formula,  $v_{id}$  and  $x_{id}$  are the velocity and position of the  $d$  dimension of the  $i$ th particle, respectively.  $\omega$  is the inertia weight;  $c_1$  and  $c_2$  are learning factors;  $r_1$  and  $r_2$  are random numbers;  $t$  is the number of iterations;  $gbest_{id}(t)$  is the  $d$ -dimensional position of the global optimal particle in the  $t$ -th iteration, and  $pbest_{id}(t)$  is the  $d$ -dimensional position of the historical optimal particle  $i$  in the  $t$ -th iteration.

By observing the second and third items on the right of Eq. (7), it can be seen that there is a 50% probability that this item will be zero, which will lead to the failure of speed updating and will not improve with the increase of iterations, significantly weakening the convergence speed and optimization ability of the algorithm. Therefore, the coding method of “dual structure coding of particle position” proposed in the literature<sup>16</sup> is introduced in this paper to construct successive particle swarm optimization (SPSO) for fault location of the distribution network, which is defined as follows:

$$v_{id}(t+1) = \omega v_{id}(t) + c_1 r_1 (\overline{pbest}_{id}(t) - \overline{x}_{id}(t)) + c_2 r_2 (\overline{gbest}_{id}(t) - \overline{x}_{id}(t)) \quad (10)$$

$$\overline{x}_{id}(t+1) = \overline{x}_{id}(t) + v_{id}(t+1) \quad (11)$$

$$x_{id}(t+1) = \begin{cases} 1, & 0.5 < \text{sign}(\overline{x}_{id}(t+1)) \\ 0, & \text{other} \end{cases} \quad (12)$$

where  $\text{sign}(x) = \frac{1}{1+e^{-x}}$ ;  $v_{id}(t) \in [-4.5, 4.5]$ ;  $\overline{gbest}_{id}(t)$ ,  $\overline{pbest}_{id}(t)$ ,  $\overline{x}_{id}(t) \in [-5k, 5k]$ . this article takes  $k = 1.5$ . The improved algorithm retains the advantages of the PSO algorithm in continuous space search and applies to discrete space optimization problems.

### Binary fish swarm algorithm

In a piece of water, fish can often find the place with more nutrients by themselves or with other fish, so the place with the most significant number of fish surviving is generally the place with the most nutrients in the water. The artificial fish swarm algorithm is based on this feature to achieve optimization. Artificial fish shoals are usually abstracted into four behaviors: swarm behavior, tailgating behavior, and foraging behavior. In recent years, the AF algorithm has become one that can solve binary optimization problems. In order to reflect the advantages and effectiveness of the algorithm established in this paper, the AF algorithm is introduced as a comparison, and its relevant definitions are as follows:

#### The gathering behavior

Fish naturally gather in groups to ensure their survival and avoid hazards during swimming. There are three rules for fish to abide by when they gather in groups: separation rules: try to avoid overcrowding with neighboring partners; Alignment rules: try to be consistent with the average direction of neighboring partners; Cohesion rule: Try to move toward the center of the neighboring partner.

Suppose the current state of  $i$  artificial fish is  $x_i$ , the food concentration is  $f_i$ , and the number of artificial fish in its field of vision ( $\text{norm}(x_i - x_j) < \text{visual}$ ) centering on itself is  $n_s$ . If  $n_s \geq 1$ , the binary central expression is:

$$x_j = \begin{cases} 1, & \rho > 0.5 \\ \text{round}(\text{rand}), & \rho = 0.5 \\ 0, & \rho < 0.5 \end{cases} \quad (13)$$

where  $\rho = \frac{\text{sum}(x_j)}{n_j}$ , round is the integer function.

Calculate the food concentration  $f_j$  of the center, if satisfied:

$$f_j < f_i, f_j/n_j < \sigma \cdot f_i \quad (14)$$

Note the center is not too crowded. Follow the formula (15) to move toward the center. Otherwise, foraging behavior is performed.

$$x_{(i+1)k} = \begin{cases} x_{jk}, & \text{select } nn \leq \text{Step positions on } x_j \text{ at random} \\ x_{ik}, & \text{other} \end{cases} \quad (15)$$

In this paper,  $n$  is obtained by the function  $\text{randperm}()$ . For Eq. (15),  $\text{randperm}(\text{Step}, 1)$  is taken here. The  $\text{Step}$  is the step size of each move.

#### The rear-ending behavior

When one or more of the fish in the shoal finds food, its neighbor follows it quickly to the spot.

Suppose the current state of artificial fish  $i$  is  $x_i$ , the food concentration is  $f_i$ , and  $x_j$  is the partner with the lowest food concentration among all partners in its field of vision centered on itself. If  $f_j > f_i$ , foraging behavior

is performed; Otherwise, use Formula (14) centered on  $x_i$  to judge whether it is crowded. If not, move with function (15); otherwise, foraging behavior will be performed.

#### The foraging behavior

Fish generally swim freely in the water at random, and when they find food, they swim quickly toward increasing food.

Set the current state of the  $i$ th artificial fish as  $x_i$  and food concentration as  $f_i$ , and select a state  $x_j$  randomly in the field of vision according to Formula (16) with food concentration as  $f_j$ .

$$x_{jk} = \begin{cases} \sim x_{ik}, & \text{select } nn < \text{Visual positions on } x_i \text{ at random} \\ x_{ik}, & \text{other} \end{cases} \quad (16)$$

If  $f_j < f_i$ , move one step in this direction according to (17). The Try-number is repeated if the maximum search times cannot be met. If the maximum search times are not met, random behavior (18) is executed

$$x_{(i+1)k} = \begin{cases} x_{jk}, & \text{randomly select } m (m \leq \text{Step}) \text{ elements from the selected } n \text{ positions} \\ x_{ik}, & \text{other} \end{cases} \quad (17)$$

#### The random behavior

Individual fish swim randomly in the water to find food spots or partners around them in a broader area.

$$x_{(i+1)k} = \begin{cases} x_{ik}, & \text{select } l (l \leq \text{Step}) \text{ positions on } x_i \text{ at random} \\ x_{ik}, & \text{other} \end{cases} \quad (18)$$

### PSO-GA and SPSO-GA algorithms for distribution network fault location

In this paper, the concepts of gene crossover and mutation in GA were introduced in the iteration of the PSO algorithm to increase the diversity of particles<sup>17</sup>. In this paper, the mutation effect is achieved by inverting a specific number, different from the variation in the previous binary genetic algorithm. In order to balance the computational efficiency, each iteration has a 20% probability of going through an additional the GA substep. After entering GA substep, iteration  $G = 30$  times, where crossover probability  $pc = 0.9$ , mutation probability  $pm = 0.5$ . When the GA substep gets a better individual than the global optimal of the previous generation, exit the GA substep immediately to improve the operation efficiency. The process is shown in Fig. 1.

### Comparison of fault optimization in SPSO-GA, PSO-GA and AF distribution networks

In order to compare and study the optimization ability of the three algorithms, this paper takes the IEEE33-node distribution network as the object. It uses MATLAB simulation to verify the algorithm's effectiveness in terms of optimization accuracy, fault tolerance, and optimization speed. The network structure of the distribution network is shown in Fig. 2:

The number of individuals of the three algorithms is set as  $N = 20$ , the number of single iterations of the program is set as 100, and each algorithm is tested 200 times to reduce occasionality.

#### Optimal parameter verification

In order to make the comparison results more convincing, it is necessary to find out the optimal parameters of each algorithm. In this paper, Section "Conclusion", the fault of the distribution network is selected as the optimization object of the three algorithms, and the grid method is used to find the optimal parameters.

The learning factors  $c1$  and  $c2$  of the particle swarm optimization algorithm are generally set in the interval of  $[0,4]$ , and the algorithm is more stable when the inertia weight  $\omega$  is less than 1. In this paper, the parameters of the PSO-GA algorithm are set as  $\omega = 0.9$ ,  $c1 = 3.4$ ,  $c2 = 3.5$ .

For the SPSO-GA algorithm proposed in this paper, The debugging steps are as follows:

- (1) This algorithm uses a different continuous velocity (only containing 0 and 1) for iteration, so the velocity of some particles will gradually approach 0, resulting in a slower position update. Therefore, the random velocity between  $(0, \pm 2)$  is re-assigned to the particles with lower velocity to retain the positive and negative of the original velocity to prevent the particles from falling into the local optimal state.
- (2) Through debugging, the basic parameters of the SPSO-GA algorithm are set as  $\omega = 0.8$ ,  $c1 = 0.7$ ,  $c2 = 0.1$ .

For the AF algorithm, Step and Visual are restricted to integers in the interval  $(0,4]$ , and the size of the delta is also restricted to the interval  $(0,4]$ . The parameter combination with the highest accuracy was selected, and finally, the parameters of the algorithm were set as Step = 2, visual = 4 and, delta = 2.5.

#### Comparison of search accuracy

Table 1 compares the positioning accuracy obtained for each optimization-seeking algorithm across 200 experiments conducted under various human-specified fault scenarios. It can be seen from the table that the optimization accuracy of the PSO-GA algorithm and AF algorithm in different fault segments changes significantly,

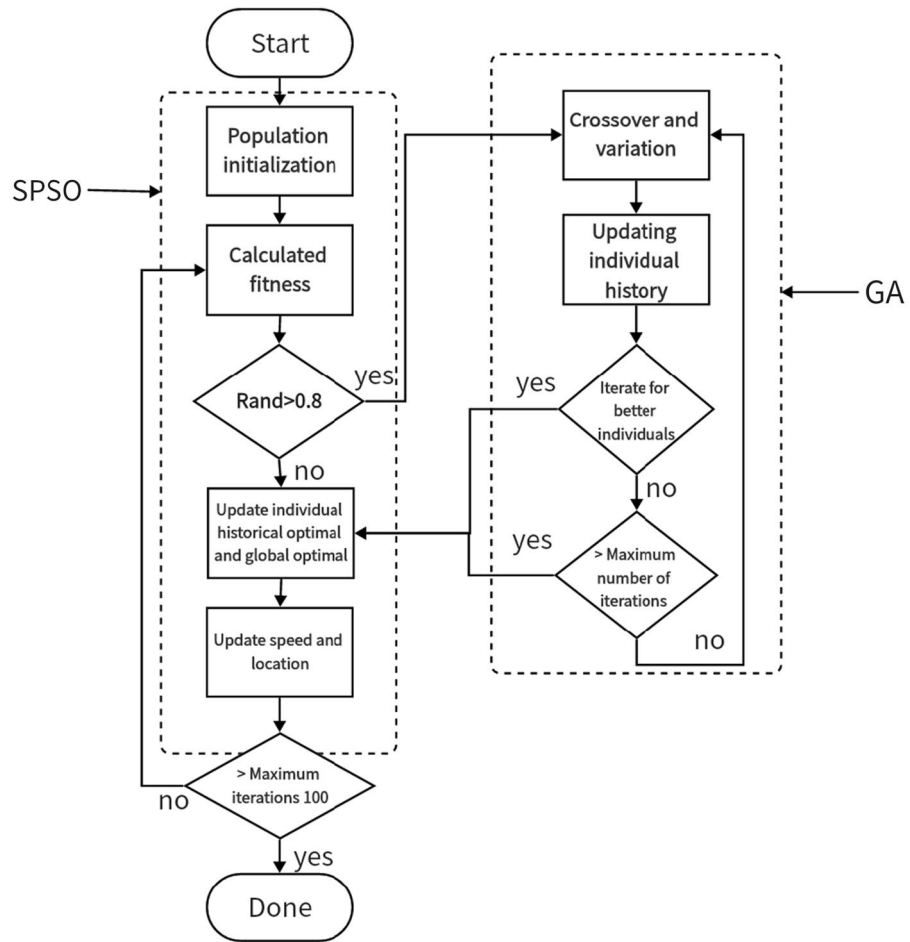


Figure 1. PSO-GA algorithm flowchart.

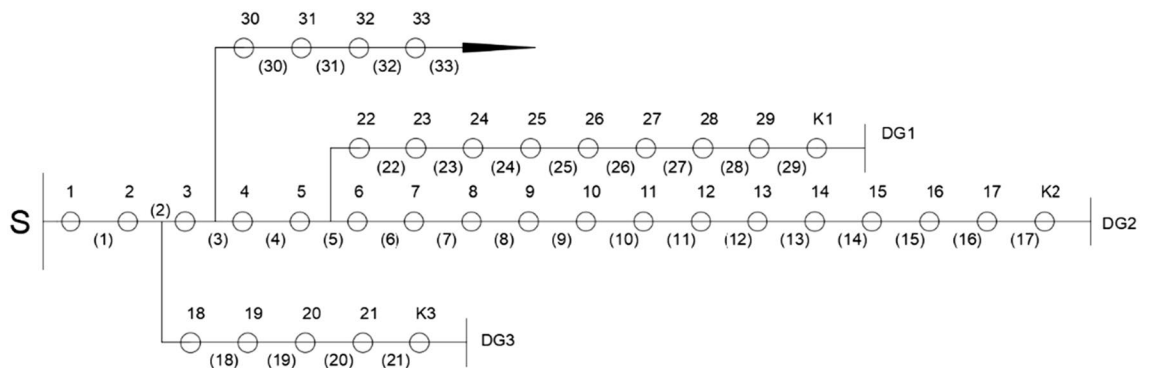


Figure 2. IEEE 33-node distribution network.

Algorithms	2 (%)	3 (%)	5 (%)	16 (%)	19 (%)	27 (%)	32 (%)
PSO-GA	100	92	95	98	11	70	16
AF	100	100	100	100	0	100	1
SPSO-GA	100	99	98.5	100	85	87	77.5

Table 1. Comparison of single fault location accuracy.

and the algorithm is unstable. At the same time, the algorithm proposed in this paper is stable and has a good optimization effect.

It can also be seen from Table 1 that when the fault occurs in section 32 of the distribution network, the optimization accuracy of the three algorithms is low; this is because there are only 20 optimization individuals set in this paper, and the probability of hybrid algorithm entering GA subroutine is only 0.2, so if we want to increase the optimization accuracy of the algorithm, in addition to increasing the number of particles in the algorithm, it can also increase the number of iterations of the GA subroutine and or increase the probability of entering the GA subroutine. Through verification, when the number of individual particles of each algorithm is increased to 30, the accuracy of the SPSO-GA algorithm constructed in this paper can reach 88%. In contrast, the optimization effect of the other two algorithms is not improved much. Figure 3 shows the iteration situation where only the particle number of each algorithm is increased to 30:

Continue to compare other fault situations. The following simulation results are still based on the particle number of the algorithm being 20, and the comparison results of dual fault location accuracy are shown in Table 2.

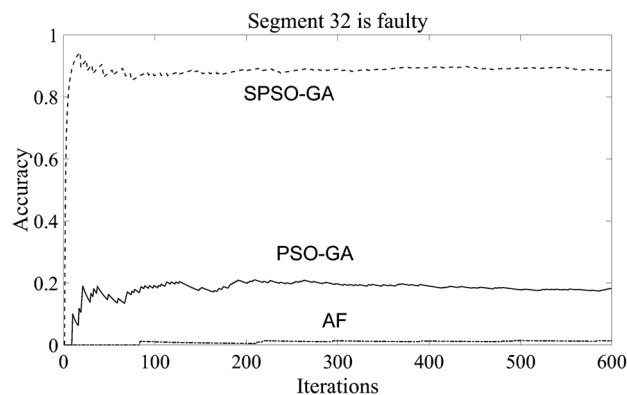
It can be seen from the optimization results of double faults that, except for the case of feeder segment 23 and 30 faults, all algorithms can maintain a high level in terms of stability and positioning accuracy, and the SPSO-GA algorithm has little difference with the AF algorithm in the optimization of double faults. However, the SPSO-GA algorithm is more stable in general.

Table 3 shows the optimization results of triple faults. It can be seen that when the faults co-occur in sections 15, 28, and 3, the optimization result of the AF algorithm drops significantly to only 68.5%. Although PSO-GA has a higher optimization rate in this case, the SPSO-GA algorithm shows a more stable optimization result.

### Comparison of fault tolerance of algorithm optimization

As the distribution network at the end of the power system, most of its faults are single faults<sup>18</sup>, so in order to highlight the fault tolerance of the SPSO-GA algorithm, this paper continues to simulate and compare single faults containing distorted information, and the calculation results are shown in Table 4.

As seen from Table 4, the optimization accuracy of the three algorithms has a high accuracy when the fault occurs in section 16, and there is one or two distorted information. Among them, the fish swarm algorithm AF



**Figure 3.** Iterative comparison after increasing the number of particles.

Algorithms	33 4 (%)	2 28 (%)	8 18 (%)	2 9 (%)	9 25 (%)	23 30 (%)	20 7 (%)
PSO-GA	91	100	92.5	100	96.5	26.5	85.5
AF	100	100	100	100	100	19.5	100
SPSO-GA	99	99.5	97.5	99.5	98	79.5	97.5

**Table 2.** Comparison of double fault location accuracy.

Algorithms	30 6 19 (%)	20 4 27 (%)	7 5 30 (%)	1 18 4 (%)	31 26 9 (%)	22 17 19 (%)	15 28 3 (%)
PSO-GA	86.5	97.5	93.5	100	97	100	88
AF	100	100	98	100	100	100	68.5
SPSO-GA	91.5	94	92.5	100	96	100	90.5

**Table 3.** Comparison of accuracy of triple fault location.

Algorithms	32(13) (%)	32(13 19) (%)	16(5) (%)	16(7 15) (%)
PSO-GA	9.5	11.5	96	97.5
AF	2	1.5	100	100
SPSO-GA	75.5	75	100	98.5

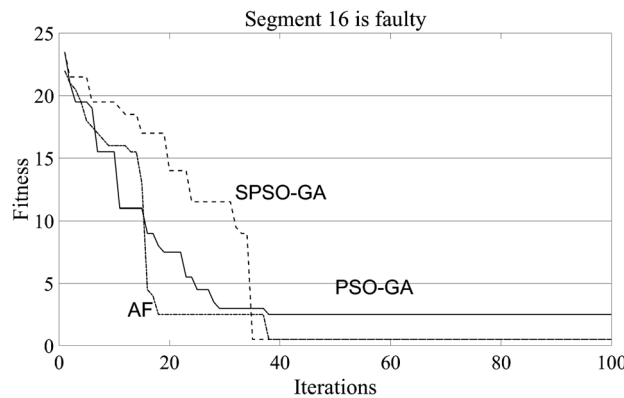
**Table 4.** Comparison of single fault location accuracy with distortion information.

can even reach 100%. However, when 32 zone faults and distorted information exist, the optimization effect of PSO-GA and AF algorithms is significantly reduced. In contrast, the SPSO-GA algorithm shows better stability despite some changes. Therefore, the SPSO-GA algorithm can better locate the fault section with distorted information and be more stable. Similarly, if the fault optimization accuracy of the algorithm needs to be improved. In that case, it can still be achieved by adding a few particles or increasing the probability of GA substeps.

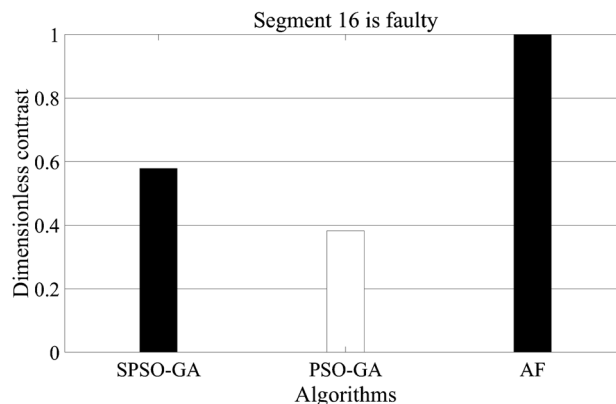
**Search for speed comparison**

Figure 4 shows the comparison results of a single iteration of the three algorithms. Since the single iteration of the algorithm is contingent, it cannot be equated with the speed of algorithm optimization. In order to compare the searching speed of each algorithm and make the comparison results reliable, fault section 16, which has little difference in the searching accuracy rate of the three algorithms in Table 1, is selected for data analysis. The specific setting of the program here is to run the three algorithms for 200 times, respectively. When the optimal value is iterated in an experiment, the experiment will be ended, and the time will be accumulated for the next experiment. The final comparison result is shown in Fig. 5.

According to the running time of the three algorithms in Fig. 5, the SPSO-GA algorithm constructed in this paper is about 1/3 longer than that of the PSO-GA algorithm. In contrast, the iteration time of the AF algorithm is more than twice as long as that of the PSO-GA algorithm. Therefore, in combination with Figs. 4 and 5, it can be concluded that under the condition of high accuracy of the three algorithms. However, AF requires fewer



**Figure 4.** Comparison of times of single iteration.



**Figure 5.** Iteration speed comparison.

iterations, and its running time is the longest among the three algorithms, which is determined by the large computation amount of the AF algorithm itself. In addition, although the iteration speed of PSO-GA is fast, combined with the previous analysis, it can be seen that the optimization accuracy of this algorithm is the lowest. Therefore, overall, the algorithm constructed in this paper is still dominant.

## Conclusion

In this paper, we break away from the binary nature of fault coding in distribution networks and introduce a successive particle swarm coding approach. This method is then combined with the principles of genetic algorithms to form a new hybrid algorithm called SPSO-GA. The accuracy, fault tolerance, and optimization speed of SPSO-GA, AF, and PSO-GA in locating faults within the IEEE33 node distribution network were compared. The results indicate that the SPSO-GA hybrid algorithm proposed in this paper exhibits high optimization accuracy, stability, and fault tolerance across various fault conditions. Moreover, it demonstrates rapid execution speed and performs well even with a small number of particles. These advantages collectively position SPSO-GA as a superior choice compared to other algorithms.

Overall, the findings of this study offer new insights into addressing fault localization issues in multi-power-source structures, contributing to reducing operational costs and improving the stability of power networks. As technology advances and algorithms are optimized, these outcomes will serve as valuable references and foundational support for achieving more reliable, intelligent, and efficient power systems.

## Data availability

The data that support the findings of this study are available on request from the corresponding author, Jiachun Li, upon reasonable request.

Received: 2 September 2023; Accepted: 3 May 2024

Published online: 17 May 2024

## References

- Bie, Z., Lin, Y., Li, G. & Li, F. Battling the extreme: A study on the power system resilience. *Proc. IEEE* **105**(7), 1253–1266. <https://doi.org/10.1109/JPROC.2017.2679040> (2017).
- Saw, B. K., Bohre, A. K., Jobanputra, J. H. & Kolhe, M. L. Solar-DG and DSTATCOM concurrent planning in reconfigured distribution system using APSO and GWO-PSO based on novel objective function. *Energies* **16**(1), 263. <https://doi.org/10.3390/en16010263> (2022).
- Zhang, T. *et al.* Restoration model for unbalanced distribution networks considering renewable sources uncertainties. *Energy Rep.* **9**, 296–303. <https://doi.org/10.1016/j.egy.2022.10.413> (2023).
- Deshmukh, B., Kumar Lal, D. & Biswal, S. A reconstruction based adaptive fault detection scheme for distribution system containing AC microgrid. *Int. J. Electr. Power Energy Syst.* **147**, 108801. <https://doi.org/10.1016/j.ijepes.2022.108801> (2023).
- Wang, Y. *et al.* A faulty line detection method for single phase-to-ground fault in resonant grounding system with CTs reversely connected. *Int. J. Electr. Power Energy Syst.* **147**, 108873. <https://doi.org/10.1016/j.ijepes.2022.108873> (2023).
- Bao, W., Fang, Q., Wang, P., Yan, W. & Pan, P. A fault location method for active distribution network with DGs. In *2021 6th International Conference on Smart Grid and Electrical Automation (ICSGEA)* 6–10. <https://doi.org/10.1109/ICSGEA53208.2021.00010> (2021).
- Wang, X. *et al.* Grounding fault location method of overhead line based on dual-axis magnetic field trajectory. *Prot. Control Mod. Power Syst.* **8**(1), 4. <https://doi.org/10.1186/s41601-023-00276-z> (2023).
- Wu, Z. *et al.* Optimal PMU Placement considering load loss and relaying in distribution networks. *IEEE Access* **6**, 33645–33653. <https://doi.org/10.1109/ACCESS.2018.2841891> (2018).
- Shan, Y., Hu, J. & Liu, H. A Holistic power management strategy of microgrids based on model predictive control and particle swarm optimization. *IEEE Trans. Ind. Inform.* **18**(8), 5115–5126. <https://doi.org/10.1109/TII.2021.3123532> (2022).
- Ou, S., Feng, B., Qin, L., Yu, X. & Luo, Y. Fault location analysis in active distribution network based on multi-population genetic algorithm. *J. Phys.: Conf. Ser.* **2237**(1), 012020. <https://doi.org/10.1088/1742-6596/2237/1/012020> (2022).
- Li, D., Luo, Y., Zhou, S., Yi, X., Xiong, Y. & Wu, Y. Bi-level optimal dispatch method of active distribution network based on improved artificial fish swarm algorithm. In *2023 6th International Conference on Energy, Electrical and Power Engineering (CEEPE)* 561–567. <https://doi.org/10.1109/CEEPE58418.2023.10166727> (2023).
- He, X. Research on fault location method of distribution network. *IOP Conf. Ser.: Earth Environ. Sci.* **647**(1), 012049. <https://doi.org/10.1088/1755-1315/647/1/012049> (2021).
- Gong, M., Xu, Z., Xie, Y., Pan, J. & Li, R. Fault-section location of distribution network containing distributed generation based on the multiple-population genetic algorithm of chaotic optimization. In *2017 Chinese Automation Congress (CAC)* 4984–4988. <https://doi.org/10.1109/CAC.2017.8243663> (2017).
- Lei, J., Guo, Y., Luo, D., Xu, Z. & Wang, R. Fault location of distribution network based on multi-population particle swarm optimization algorithm. *J. Phys.: Conf. Ser.* **2360**(1), 012024. <https://doi.org/10.1088/1742-6596/2360/1/012024> (2022).
- Ying, W., Zhan, S., Zhongmiao, K. & Xingnan, L. Research and simulation of IBPSO-based fault location in power distribution network adopting DG. *IOP Conf. Ser.: Earth Environ. Sci.* **242**, 022027. <https://doi.org/10.1088/1755-1315/242/2/022027> (2019).
- Macur, E. & Kiraz, B. Opposition based binary particle swarm optimization algorithm for feature selection. In *2022 Innovations in Intelligent Systems and Applications Conference (ASYU)* 1–6. <https://doi.org/10.1109/ASYU56188.2022.9925378> (2022).
- Shi, L., Gong, J. & Zhai, C. Application of a hybrid PSO-GA optimization algorithm in determining pyrolysis kinetics of biomass. *Fuel* **323**, 124344. <https://doi.org/10.1016/j.fuel.2022.124344> (2022).
- Liu, K., Zhang, C., Chen, W. & Zhao, J. Fault location method combining transient mutation and dynamic time warping algorithm for grounding system. *Energy Rep.* **9**, 37–45. <https://doi.org/10.1016/j.egy.2022.10.362> (2023).

## Acknowledgements

At the end of this paper, I would like to express my special thanks to Associate Professor Yang Lu and Professor Jiachun Li for their guidance and help in writing, revising and finishing the paper. At the same time, I would like to thank the key Technology research and development Program of Guizhou Province (Grants No. QKHZC [2021] YB368) for providing necessary financial support for the completion of this research, so that the research



can be carried out smoothly. Finally, I would like to express my heartfelt thanks to my family and friends for their support and companionship in my life and study.

### Author contributions

W.X.: Conceptualization, Formal analysis, Software, Writing-original draft. J.L.: Formal analysis, Validation. L.Y.: Conceptualization, Funding acquisition, Resources, Supervision, Writing-review & editing. Q.Y.: Supervision, Funding acquisition, Writing-review & editing.

### Funding

This work was supported by the key Technology research and development Program of Guizhou Province.

### Competing interests

The authors declare no competing interests.

### Additional information

**Correspondence** and requests for materials should be addressed to J.L. or L.Y.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024