

## Article

# Fault Recovery of Distribution Network with Distributed Generation Based on Pigeon-Inspired Optimization Algorithm

Mingyang Liu <sup>1</sup>, Jiahui Wu <sup>1,2,\*</sup>, Qiang Zhang <sup>3</sup> and Hongjuan Zheng <sup>4</sup><sup>1</sup> School of Electrical Engineering, Xinjiang University, Urumqi 830017, China; 107552104065@stu.xju.edu.cn<sup>2</sup> Engineering Research Center of Renewable Energy Power Generation and Grid Connection Control, Ministry of Education, Urumqi 830017, China<sup>3</sup> State Grid Xinjiang Integrated Energy Service Co., Ltd., Urumqi 830000, China; zhangqiangxj123@sina.com<sup>4</sup> Nari Technology Co., Ltd., Nanjing 211106, China; zhenghj26@sina.com

\* Correspondence: wjh229@xju.edu.cn

**Abstract:** In this paper, a fault recovery strategy for a distribution network based on a pigeon-inspired optimization (PIO) algorithm is proposed to improve the recoverability of the network considering the increased proportion of distributed energy resources. First, an improved Kruskal algorithm-based island partitioning scheme is proposed considering the electrical distance and important load level during the island partitioning process. Secondly, a mathematical model of fault recovery is established with the objectives of reducing active power losses and minimizing the number of switching actions. The conventional PIO algorithm is improved using chaos, reverse strategy, and Cauchy perturbation strategy, and the improved pigeon-inspired optimization (IPIO) algorithm is applied to solve the problem of fault recovery of the distribution network. Finally, simulation analysis is carried out to verify the effectiveness of the proposed PIO algorithm considering a network restoration problem after fault. The results show that compared with traditional algorithms, the proposed PIO algorithm has stronger global search capability, effectively improving the node voltage after restoration and reducing circuit loss.

**Keywords:** Cauchy perturbation; chaos and reverse strategies; distribution network fault reconstruction; islanding detection; pigeon-inspired optimization (PIO)



**Citation:** Liu, M.; Wu, J.; Zhang, Q.; Zheng, H. Fault Recovery of Distribution Network with Distributed Generation Based on Pigeon-Inspired Optimization Algorithm. *Electronics* **2024**, *13*, 886. <https://doi.org/10.3390/electronics13050886>

Academic Editors: Akhtar Kalam and Seyed Morteza Alizadeh

Received: 4 January 2024

Revised: 21 February 2024

Accepted: 23 February 2024

Published: 26 February 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

This paper introduces, for the first time, a fault restoration and restoration strategy for distribution networks with distributed generators based on the PIO algorithm. In applying the PIO algorithm, higher randomness and wider coverage of the solution space for the initial pigeon positions are generated by incorporating chaos [1,2] and reverse strategies [3]. An elite selection strategy is employed to make the initial values closer to the global optimal solution. Simultaneously, to counteract the drawback of the conventional PIO algorithm that tends to get trapped in local optima, a Cauchy perturbation and redistribution strategy [4,5] is introduced during the iterative process. This strategy disturbs the global optimal solution when the variable *FC* (failure counter) exceeds the predetermined threshold *MFC* (maximum failure counter), preventing premature convergence during the fault restoration process. Firstly, the *FC* variable, initialized to 0, is introduced to monitor the changes in fitness values during each iteration. When the fitness values remain relatively unchanged within a generation, it indicates premature convergence of the population, and the *FC* is increased by one. However, Cauchy perturbation should not be immediately applied at this point. This prevents misinterpreting the temporarily obtained suboptimal positions as premature convergence, especially when the population is still converging towards the global optimum. Secondly, a threshold *MFC* is set. When the *FC* surpasses the *MFC*, it is more likely that the population is trapped in a local optimum. At this stage, the Cauchy perturbation redistribution strategy can be introduced to perturb the

current global best position. As a result, half of the population that is under-performing is redistributed to escape from local optima. Simulation studies demonstrate the effectiveness of the proposed algorithm and the practicality of the restoration strategy.

The distribution network, being an essential component of the power system, plays a critical role in transmitting electrical energy from transformers to households. With the increasing demand for energy and a growing emphasis on environmental sustainability, the distribution network, as a key part of the power system, is facing new challenges and opportunities [6,7].

Among the various research areas in distribution networks, faults in distribution networks directly impact societal productivity and the daily power supply needs of citizens. As the integration of renewable energy sources into the distribution network increases, traditional radial distribution networks are gradually transforming into systems of multi-source and multi-sections. This structural evolution leads to fundamental changes in the network's configuration and power flow. As a result, the complexity of fault within distribution networks is progressively rising, which adversely affects the operation and reliability of the power system [8]. In contrast to protection strategies in conventional systems during network faults, topology alteration through switching operations [9] is commonly employed in the new networks to isolate faulty sections [10,11], ensure power supply to non-faulty zones, optimize the network's structure during fault periods, and incorporate distributed energy resources for islanded partition [12,13]. This effectively supplies power to users in islanded segments before fault restoration, thereby forming a highly efficient fault recovery approach.

Both fault restoration and islanding partition fundamentally achieve fault recovery by altering the topology characteristics of distribution networks [14]. Scholars worldwide have proposed various solutions to address these challenges. In [15], an improved ant colony algorithm is presented by randomizing the initial selection of the first branch and eliminating common heuristic values, which avoids suboptimal solutions when solving the distribution network restoration problem. However, this algorithm relies on the random movement of ants in the solution space, and its optimization efficiency is limited. In [16], adaptive crossover genetic mutation operators are introduced into the genetic algorithm, achieving promising outcomes in the context of problem enhancement. On the other hand, genetic algorithms typically require significant computational resources during iterations, especially for complex problems. The computational cost can be high, and they do not guarantee convergence to global optimum in all cases. A hybrid particle swarm algorithm that combines binary and discrete particle swarm methods is proposed for solving distribution network restoration problems [17]. However, the requirements of parameter tuning and debugging make the algorithm less straightforward to use in practical applications. In [18], a combination of the chaos algorithm and immune algorithm is utilized, suggesting a chaos-immune algorithm for solving the restoration problem. Additionally, the chaos algorithm is sensitive to initial conditions, which significantly impact the results and introduces a degree of uncertainty. In [19,20], the restoration problem is addressed through a combination of heuristic rules and intelligent algorithms; however, integration of intelligent algorithms leads to a prolonged computation time and faces uncertainties in achieving global optimal solutions.

In summary, limitations of the methods mentioned in the literature for distribution network fault restoration include local search issues, high computational costs, complex parameter tuning, and uncertainty. To address the aforementioned limitations, an improved pigeon-inspired optimization (PIO) algorithm is employed in this paper. The improved pigeon-inspired optimization (IPIO) algorithm has advantages of global search capability, diverse initial populations, and Cauchy perturbation strategy. It outperforms ant colony algorithms, genetic algorithms, and hybrid particle swarm algorithms in terms of performance. Although it may require more iterations, it experiences fewer instances of getting stuck in local optima and has a higher likelihood of finding the optimal solution. It is

expected that this enhanced PIO algorithm will overcome these limitations and provide more feasible solutions for distribution network fault restoration problems.

The PIO was introduced in 2014 by Duan Haibin et al. [21], drawing inspiration from pigeons’ unique navigation abilities. The PIO algorithm initially found applications in cooperative multi-drone systems and aerospace engineering [22]. Compared to traditional optimization algorithms like genetic algorithms and ant colony algorithms, the PIO algorithm employs two distinct operator models to simulate different navigation tools used by pigeons during various stages of their homing journey. Therefore, the PIO exhibits characteristics such as fast convergence speed and high search efficiency when seeking global optimal solutions.

## 2. Restoration Mathematical Model

### 2.1. Objective Function

During distribution network faults, distributed generators (DGs) have capabilities to act as independent power supply units which remain connected to the main grid after the fault. DGs that are in communication with the control dispatch center are used as “black start” power sources [23]. They are switched to isolated island operation mode to supply surrounding loads, thereby minimizing loss of load. Fault restoration involves adjusting the status of switches to optimize the power flow in the remaining distribution network. Therefore, the aim is to find an optimal topology for the distribution network under fault conditions that facilitates a swift restoration of the power system to its normal state after the fault. The choice of different objective functions leads to varying fault recovery outcomes, which in turn affects the outcome.

The recovery objective function chosen in this paper is to minimize the active power losses in the lines and reduce the number of switching operations [24,25]. Mathematically, the objective function is defined as follows:

$$\min P_{\text{loss}} = \sum_{i=1}^n I_i^2 R_i \tag{1}$$

$$\min f_{\text{switch}} = \sum_{i=1}^{m1} (1 - B_i) + \sum_{j=1}^{m2} C_j \tag{2}$$

where  $P_{\text{loss}}$  represents the active power losses in the main grid;  $f_{\text{switch}}$  indicates the number of switching operations in the distribution network during the restoration process;  $I_i$  represents the current flowing through branch  $i$ ; and  $R_i$  denotes the resistance of branch  $i$ .  $m1$  and  $m2$  represent the numbers of switching operations in branches and tie lines in the remaining network after islanding;  $B_i$  and  $C_j$  denote the status of branch switches and tie switches, respectively, where 1 and 0 represent the closed and open states of the switches.

To transform the multi-objective restoration problem into a single-objective one for function optimization, different weight factors for the two objective functions in (1) and (2) are introduced and then normalized. The objective function for the restorative restoration problem of the distribution system can be formulated as

$$\min f = \min \left( \mu \frac{P_{\text{loss}}}{P_0} + v \frac{f_{\text{switch}}}{f_0} \right) \tag{3}$$

where  $P_0$  denotes the active power losses before restoration;  $f_0$  represents the total number of switches in the network after islanding; and  $\mu$  and  $v$  signify the weights of the single objective function and must satisfy  $\mu + v = 1$ .

### 2.2. Restoration Constraint Conditions

#### (1) Branch Flow Constraints

$$U_{j,t}^2 = U_{i,t}^2 - 2(r_{ij}P_{ij,t} + x_{ij}Q_{ij,t}) + (r_{ij}^2 + x_{ij}^2)I_{ij,t}^2 \tag{4}$$

$$p_{j,t} = P_{ij,t} - r_{ij}I_{ij,t}^2 - \sum_{k:j \rightarrow k} P_{jk,t} \quad (5)$$

$$q_{j,t} = Q_{ij,t} - x_{ij}I_{ij,t}^2 - \sum_{k:j \rightarrow k} Q_{jk,t} \quad (6)$$

$$I_{ij,t}^2 = \frac{P_{ij,t}^2 + Q_{ij,t}^2}{U_{i,t}^2} \quad (7)$$

where  $U_{i,t}$  and  $U_{j,t}$  are the voltages at nodes  $i$  and  $j$ ;  $P_{i,t}$  and  $P_{j,t}$  are the active and reactive power injections at node  $i$  and  $j$ ;  $q_{i,t}$  and  $q_{j,t}$  are the active and reactive power injections at node  $i$  and  $j$ ;  $P_{ij,t}$  and  $Q_{ij,t}$  are the active and reactive powers at the starting end of branch  $ij$ ;  $r_{ij} + jx_{ij}$  is the impedance of branch  $ij$ ; and  $P_{jk,t}$  and  $Q_{jk,t}$  are the active and reactive powers at the ending end of branch  $jk$ .

- (2) Node Voltage Constraints

$$U_{\min} \leq U_{i,t} \leq U_{\max} \quad (8)$$

where  $U_{\max}$  and  $U_{\min}$  represent the maximum and minimum allowable voltages at each node in the distribution system.

- (3) Line Power Constraints

$$S_i \leq S_{i\max} \quad (9)$$

where  $S_{i\max}$  represents the maximum power limit that the  $i$ th line can carry.

- (4) DG Output Constraints

$$P_{\text{DG},\min} \leq P_{\text{DG}} \leq P_{\text{DG},\max} \quad (10)$$

$$0 \leq Q_{\text{DG}} \leq P_{\text{DG},\max} \tan \varphi \quad (11)$$

where  $P_{\text{DG},\min}$  and  $P_{\text{DG},\max}$  represent the lower and upper limits of DG active power output and  $\varphi$  is the power factor angle.

- (5) Radial Structure Constraints

The voltages and power around the fault area should not be affected by the restoration to ensure the safety of the power system. After islanding partition, the isolated models of the system should possess a radial topology, meaning the presence of loops should be avoided.

### 3. Islanding Partition Scheme Based on Kruskal Algorithm

In the protection mechanism of the distribution network, protective devices are only installed at the beginning of the feeder line, so in case of a fault on any branch, protective overcurrent (OCR) relay will disconnect the 1st branch, rendering the entire feeder dead. Only by switching will part of the load be restored in island mode and after that the restoration will be performed.

#### 3.1. Islanding Partition Scheme

In this paper, the approach for islanding operation is as follows. Firstly, loads are divided into three levels (1 to 3) based on their importance and accordingly weights are assigned to each level.

Next, the electrical distances between various load nodes and DGs within the island are used as the criteria. Loads with shorter distances are given priority in the islanding process. Lastly, while ensuring that the total load within the island does not exceed the supply capacity of DGs, efforts are made to minimize the active power losses during the islanding operation, ensuring the island's economic performance. The islanding model can be expressed as

$$\max PL = \sum_{j=1}^m \sum_{i=1}^n \frac{\chi_i P_i}{D_{ki,j}} \quad (12)$$

where  $PL$  represents the equivalent load size;  $\chi_i$  is the weight coefficient of the load at node  $i$ ;  $P_i$  denotes the active power consumption of the load at node  $i$ ;  $D_{ki}$  represents the electrical distance between DG at node  $k$  and the load at node  $i$ , which is the impedance between them;  $m$  is the total number of islands in the entire system; and  $n$  is the total number of disconnected nodes during faults.

To ensure the power balance of islands after system faults, the power generation capacity of DGs within each island should exceed the total load within the island. Considering that reactive power compensation is concentrated on the load side in the distribution network, this constraint primarily focuses on active power. The power balance constraint is expressed as

$$\sum_{j=1}^m P_{DGj} - \sum_{i=1}^n P_i - P' > 0 \tag{13}$$

where  $P_{DGj}$  represents the power generation of the  $j$ th DG within the island;  $P_i$  represents the active power of the  $i$ th load within the island;  $P'$  represents the active power losses in the lines of island;  $m$  is the number of DGs within the island; and  $n$  is the number of load points within the island.

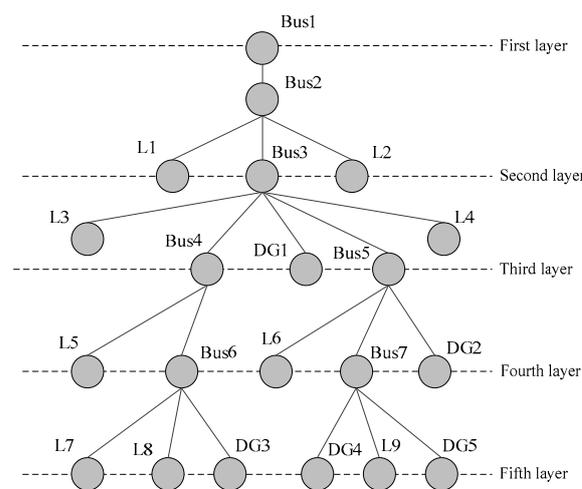
Distribution Line Safety Constraint.

$$U_{\min} \leq U_i \leq U_{\max} \tag{14}$$

$$I_{ij} \leq I_{ij}^{\max} \tag{15}$$

where  $U_{\max}$  represents the maximum allowable voltage at each node within the island;  $U_{\min}$  represents the minimum allowable voltage at each node within the island; and  $I_{ij}^{\max}$  represents the maximum current that a line within the island can withstand.

Under normal operating conditions, the network topology of a distribution system adopts a radial structure. If a ring structure is adapted, the distribution network typically operates in an open-loop configuration. Therefore, the distribution network can be visualized as a tree with the system measurement points as the root nodes and various loads as leaf nodes. Integration of DGs does not alter the network's topology; instead, they can be considered as leaf nodes with power source attributes. A connected distribution network diagram with DGs is illustrated in Figure 1.



**Figure 1.** Connection diagram of distribution network with DG.

### 3.2. The Kruskal Algorithm and the Island Partitioning Process

The Kruskal algorithm is a greedy approach employed to find the Minimum Spanning Tree (MST) of a graph. It is suitable for solving the MST of sparsely connected graphs [26]. The key idea of this algorithm is to systematically select edges starting from the smallest

ones while ensuring that the selected edges do not form loops, thus constructing a minimum spanning tree or performing island partitioning. While dividing the distribution networks into islands, this approach can help system operators to better understand the network's structure, improve response time to faults, and take appropriate measures to maintain the reliability of the power system. Considering various constraints in the distribution network, directly applying the Kruskal algorithm might not yield unique results during the search process. Additional steps are required to adjust the search conditions and orders, which is conducted by incorporating a validation process to ensure compliance with various constraints of the distribution network. This adaptation makes the Kruskal algorithm more suitable for solving the islanding problem of distribution networks with DGs.

The tree composed of a connected graph is represented in Figure 1.  $V_i$  denotes the vertex of the connected graph, and  $E_i$  denotes the edge. The generated tree is defined by a binary tuple as  $T = (V, E)$ , where  $E_i = \{(V_i, V_j), W_{ij}\}$  and  $W_{ij}$  represents the weight or impedance magnitude of the edge connecting  $V_i$  and  $V_j$ . The specific steps of this algorithm are outlined as follows:

- (1) Initialize the distribution network by assigning appropriate weight coefficients to different loads and partitioning islands with DGs possessing independent power supply capabilities.
- (2) Gradually assign loads to the islanded models and verify the power balance constraints for each load inclusion. This process maximizes the added loads while adhering to the constraints. Finally, iterate through all islands based on island numbering.
- (3) Adhere to transmission line safety constraints by assigning allocated loads to islands with available load capacity, based on the minimum electrical distance.
- (4) Optimize MST by removing bus nodes that are included in the islanded partition but are not connected to loads from the tree, resulting in the desired islanding partition scheme represented by the tree  $T = (V, E)$ .

#### 4. Fault Restoration of Distribution Networks Based on Pigeon-Inspired Algorithm

##### 4.1. Basic Pigeon-Inspired Algorithm (PIO)

This algorithm simulates two operator models used in the algorithm based on the different navigation tools used by pigeons on their journey home: (i) the Map and Compass Operator, and (ii) the Landmark Operator.

Map and Compass Operator.

$$V_i(t) = V_i(t-1) \times e^{-Rt} + \text{rand} \times (X_g - X_i(t-1)) \quad (16)$$

$$X_i(t) = X_i(t-1) + V_i(t) \quad (17)$$

where  $R$  represents the map and compass factor, which ranges between 0 and 1; rand are random numbers between 0 and 1;  $X_g$  denotes the current global best position; and  $t$  represents the generation number.

In a two-dimensional space, the velocity of the  $i$ th pigeon is determined by its previous velocity and the current position relative to the current best position of the pigeon flock. Similarly, the position of the  $i$ th pigeon depends on its previous position and its current velocity. By comparing the positions of all pigeons, the individual with the best fitness value is selected as the current global optimal position,  $X_g$ .

Landmark Operator.

$$Np(t) = \frac{Np(t-1)}{2} \quad (18)$$

$$X_c(t) = \frac{\sum X_i(t) \times \text{fitness}(X_i(t))}{Np(t) \times \sum \text{fitness}(X_i(t))} \quad (19)$$

$$X_i(t) = X_i(t-1) + \text{rand} \times (X_c(t) - X_i(t-1)) \quad (20)$$

where  $\text{fitness}()$  represents the fitness of each pigeon, indicating the quality of the solution after evaluation;  $Np$  signifies the number of pigeons remaining in each generation, which amounts to half of the original population;  $X_c(t)$  is the central position of all pigeons in generation  $t$ ; and  $\text{rand}$  is a random number between 0 and 1.

In the landmark operator evaluation phase, the navigation reference direction for the pigeon flock is based on the center position of superior individuals (pigeons familiar with landmarks). During this phase, pigeons will no longer have the interference from their individual velocity. As a result, the population can quickly converge to the optimal value.

#### 4.2. Improved Pigeon-Inspired Optimization Algorithm (IPIO)

Although the PIO algorithm exhibits fast convergence, it has its own drawbacks. When the initial population is densely distributed or does not lie near the true global optimum, the algorithm might move in the wrong direction, converging to a local optima. Moreover, the conventional PIO algorithm eliminates half of the pigeons with lower fitness in each iteration, resulting in reduced population diversity after a few iterations. Consequently, the solution space becomes limited and increases the likelihood of falling into local optima.

To address these limitations, a “chaos and reverse” strategy is introduced during pigeon flock initialization, which enhances the randomness of the generated initial solutions and widens the coverage of the solution space. Additionally, in the iteration process, a variable called failure count ( $FC$ ) is introduced. When the value of  $FC$  exceeds a predefined threshold,  $MFC$ , a Cauchy perturbation redistribution strategy is employed to disturb the global best solution, thereby preventing premature convergence.

##### 4.2.1. Chaos and Reverse Strategy

Chaos motion, driven by its inherent patterns, can traverse each trajectory within a certain range without repetition or crossing itself [1]. By utilizing chaos motion, a more diversified initial pigeon flock can be generated, effectively preventing premature convergence during optimization. Two common methods for generating chaotic variables include (i) the Tent Map method, and (ii) Logistic method. Among these methods, the Tent Map method offers a broader coverage of the solution space and faster iteration speed [27,28]. The Tent Map mapping is defined as

$$X_{n+1} = \mu \times (1 - 2 \times |X_n - 0.5|)n = 0, 1, 2, \dots, N \quad (21)$$

In (21),  $0 < X_0 < 1$ , and its value is generated using the  $\text{rand}$  function. By selecting  $\mu = 1$ , the mapping remains in a completely chaotic state, and  $X_n \in (0, 1)$ . Utilizing this method, random numbers between 0 and 1 are generated. Subsequently, based on the solution space  $[S_{\min}, S_{\max}]$  of the problem at hand and the dimension  $D$  of the PIO algorithm, the corresponding individual initialization Formula (21) can be transformed into

$$S_{i,j} = S_{\min,j} + X_n \times (S_{\max,j} - S_{\min,j})j = 1, 2, \dots, D \quad (22)$$

where  $S_{i,j}$  represents the  $j$ th dimension variable of the  $i$ th individual in the population;  $S_{\min}$  and  $S_{\max}$  represent the lower and upper bounds of the solution space, respectively.

Although using the chaos method enhances the diversity of the initial population, there remains a certain degree of randomness, and the coverage of the solution space might not be optimal. Therefore, a reverse strategy is implemented to enhance the initialization process further, with an aim to bring the initial values closer to the global optimum. The concept of reverse learning was proposed by Professor Tizhoosh in 2005, based on which utilizing both the current solution and the reverse solution as starting points during the search process can effectively enhance the algorithm’s efficiency. Thus, an elite selection strategy is employed based on the current solution and the reverse solution to generate an initial pigeon flock closer to the global optimum [29,30]. The definition of the reverse point is:

In the D-dimensional space of the algorithm,  $S = (X_1, X_2, \dots, X_D)$  is used to represent a point in space. Here,  $X_1, X_2, \dots, X_D \in R$  and  $X_i \in [a_i, b_i] \forall i \in \{1, 2, \dots, D\}$ . As for the reverse point  $\hat{S} = (\hat{X}_1, \hat{X}_2, \dots, \hat{X}_D)$  corresponds to each space point, which is expressed as

$$\hat{X}_i = a_i + b_i - X_i \tag{23}$$

For the initial population generated through chaotic motion, the reverse point  $\hat{S}$  corresponding to each individual  $S$  is computed. These reverse points are then substituted in the fitness function,  $f(\Delta)$ , to calculate the fitness values of each individual. A smaller fitness value indicates a superior solution. Each individual's fitness is compared, and if  $f(\hat{S}) \leq f(S)$  is smaller than  $S$ , the individual is replaced by its reverse point  $\hat{S}$ ; otherwise, the original individual remains unchanged.

The quality of initial solution is enhanced by iteratively comparing the fitness values of the initialized points with their corresponding "symmetric points" in the solution space and combining the same with the elite selection strategy.

#### 4.2.2. Cauchy Perturbation Redistribution Strategy

In the subsequent iteration stages of the conventional PIO algorithm, the population tends to move towards the center of better individuals during the landmark operator phase, resulting in high convergence speed. However, as the distances between pigeons decrease and most individuals gather within a certain range, their velocities decrease and the algorithm may get stuck in local optima. To overcome this limitation, a Cauchy perturbation redistribution strategy is introduced to disturb the global best solution, forcing the individuals to continue moving and making them escape from the global optima.

The Cauchy probability density function is defined as

$$f(x; x_0, \gamma) = \frac{1}{\pi\gamma \left[1 + \left(\frac{x-x_0}{\gamma}\right)^2\right]} = \frac{1}{\pi} \left[ \frac{\gamma}{(x-x_0)^2 + \gamma^2} \right] \tag{24}$$

Further, the Cauchy distribution function is defined as

$$F_t(x) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{x}{t}\right) \tag{25}$$

where  $x_0$  represents the location parameter of the Cauchy distribution;  $\gamma$  represents the scale parameter of the Cauchy distribution. Different values of  $x_0$  and  $\gamma$  correspond to different probability density functions. In this paper, values  $x_0 = 0$  and  $\gamma = 1$  are chosen to achieve significantly different function values within a small range of independent variables. Then, a Cauchy perturbation is applied to the global optimum solution.

$$X_g = X_g + C(x_0, \gamma) \times (ub - lb) \tag{26}$$

where  $X_g$  represents the global best solution;  $C(x_0, \gamma)$  represents a random number generated from the Cauchy probability distribution; and  $ub$  and  $lb$  represent the upper and lower bounds of the optimization problem for the pigeon population, respectively. The procedure involves several steps after obtaining the new  $X_g$ . As a first step, it should be checked whether the perturbed coordinate of this dimension falls within the defined bounds; if it exceeds the bounds, the coordinate should be kept unchanged. Second, if the perturbed value lies within the bounds, the fitness of the perturbed solution should be compared with that of the original global best solution. If the perturbed solution proves to be better, the original global best solution is updated with the perturbed one, and the  $FC$  is reset to 0. Finally, based on the fitness ranking, the under-performing half of the population is redistributed, converging towards the newly updated global best position.

### 4.2.3. Iteration Factor

In the landmark navigation phase of the conventional PIO algorithm, the flock moves towards the center of the current superior individual’s position in each iteration. In every generation, half of the individuals with lower fitness values are eliminated. This practice leads to a reduction in the population size by half during each initialization, even if the initial population size is large. An excessively higher elimination rate not only decreases the diversity of the population but also risks losing potential global optimal solutions in the initial stages. Moreover, a reduction in population size also contributes to a decreased convergence rate in subsequent searches.

To address these concerns, an elimination factor  $e = 0.9$  is introduced in the paper to control the elimination rate during the iteration process. This factor ensures that only 10% of the under-performing individuals are eliminated based on their fitness in each iteration. This adjustment aims to enhance the applicability of the IPIO algorithm to the fault recovery problem in distribution networks. The specific process is illustrated in Figure 2.

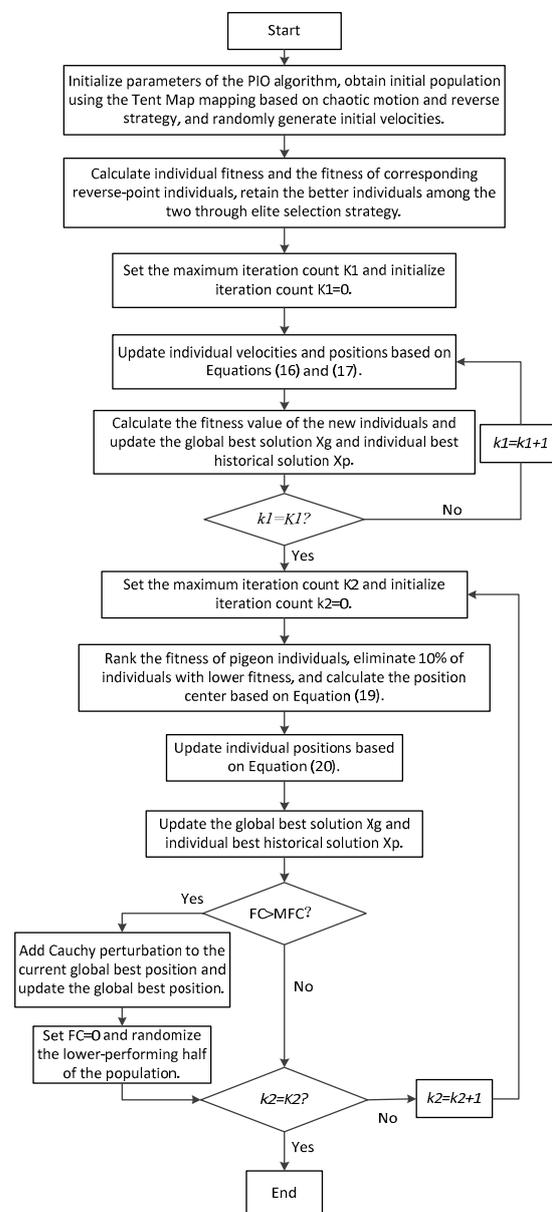


Figure 2. Improved pigeon-inspired optimization flowchart.

### 5. Simulation Verification and Analysis

#### 5.1. Basic Parameter Settings for Simulation

Using MATLAB R2021a 64 bit software for programming calculations, the computer is configured with AMD Ryzen 7 7840HS with Radeon 780M Graphics 3.80 GHz and 16.00 GB of memory. The proposed fault recovery strategy utilizing the IPIO algorithm is validated on the IEEE 33-node distribution system in terms of its reliability and effectiveness. The topological structure of the test system is depicted in Figure 3, which consists of 33 nodes and 37 branches. The base value of the voltage is 12.66 kV with a total load of 6.43187 + j2.82064 MVA. The study considers integration of four DGs, among which only DG1 lacks the ability to operate independently and cannot switch to islanded mode during faults. The specific parameters are listed in Table 1.

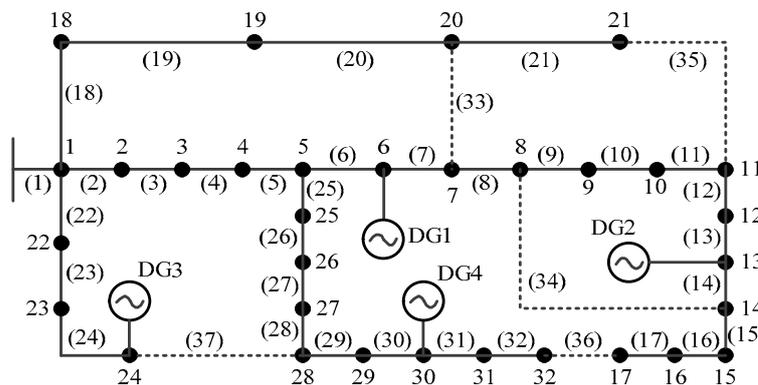


Figure 3. IEEE 33 node system containing DG.

Table 1. Parameters of the distributed resources.

DG	Location	Capacity/kW	Power Factor
1	6	700	0.85
2	13	500	0.9
3	24	1200	0.8
4	31	650	0.9

To highlight the significance of loads at various nodes in Figure 3, their load levels and corresponding weights are presented in Table 2. When a fault occurs in the distribution network, the first step involves the use of the “black-start” capability of the DGs to create islanded segments. Subsequently, the remaining system is subjected to fault recovery based on the objective function described in the paper. This process aims to enhance the node voltages during the fault periods, thereby reducing active power losses in the lines. By leveraging the IPIO algorithm, the recovery process is performed in this paper. The initial population size is set to 300, the dimension of the pigeon’s flight space is 30, and the factors for the map and compass operators are set as  $R = 0.3$ . The iteration count is defined as  $K1 = K2 = 200$ .

Table 2. Load characteristics of each node.

Load Level	Weight $\chi_i$	Node
Level 1 load	100	5, 6, 12, 13, 23, 24, 29, 31
Level 2 load	10	7, 11, 15, 22, 26, 30, 32
Level 3 load	1	1~4, 8, 9, 10, 14, 16~20, 21, 25, 27, 28

It is to be noted that DG1 is an intermittent power source and does not possess an independent power supply capability, like a wind farm or a photovoltaic system that does

not include energy storage. Specifically, DG3 does not rely on a central control system and is managed through local strategies, allowing for islanding based on the location of the fault point in the event of a distribution network failure. Considering the conditions that fulfill the islanding criteria, the initial step involves partitioning the distribution network into islands based on the capabilities of the distributed power sources. The outcome of the islanding process is presented in Table 3.

**Table 3.** Islanding results.

Number	Island Load Point	Island Load/kV	Disconnected Switch
DG2	10, 11, 12, 13, 14, 15, 16	465 kW + j230 kvar	S10, S16
DG3	22, 23, 24	930 kW + j450 kvar	S22
DG4	29, 30, 31, 32	620 kW + j810 kvar	S29

### 5.2. Restoration Simulation Results

After completing the islanding process, fault recovery is required for the remaining system that is not a part of any island. Initially, a part of the main system is completely disconnected from the islands. Then, the interconnection switches within the main system are adjusted to restore the system while satisfying the constraints. The recovery steps are illustrated in Figure 2.

To validate the feasibility and optimality of the fault recovery strategy proposed in this paper, two sets of control experiments were conducted and the results were compared. Firstly, the feasibility of the proposed IPIO algorithm in distribution network restoration was verified. It performed better compared to the conventional PIO algorithm used in unmanned aerial vehicles and aerospace applications. Secondly, the superiority of the proposed IPIO algorithm was demonstrated by comparing it with the Genetic Algorithm (GA).

Finally, the performance of the proposed algorithm was compared with other algorithms to further validate the conclusions of this paper. The system was optimized for restoration when specific nodes experience faults.

#### 5.2.1. Fault on Branch 28

The fault occurred in the upstream of DG3, which is a distributed power source that cannot communicate continuously with the central system, resulting in its shutdown. DG1 and DG2 were connected to the remaining system to assist in the recovery of the main grid. The switch S29 in the downstream of branch 28 was isolated, making DG4 operate in islanded mode. The results of the fault recovery are presented in Table 4.

**Table 4.** Results of fault recovery and restoration (Fault on Branch 28).

Algorithm	Minimum Voltage/pu	Active Power Loss/kW	Disconnected Switch Number	Switching Operations Count
Before	0.9683	49.3339	S33, S34, S35, S36, S37	--
IPIO	0.9746	42.9386	S29, S36, S7, S11, S13	8
PIO	0.9646	52.2051	S29, S36, S14, S33, S35	4
GA	0.9729	43.4675	S29, S36, S7, S34, S35	4

It is evident from the results presented in Table 4 that the proposed IPIO algorithm for solving the distribution network fault recovery problem leads to notable improvements compared to the initial state. Specifically, the active power loss in the reconstructed lines was reduced by 6.3953 kW, and the lowest node voltage in the network was increased by

0.0063 per unit (pu). These improvements outperform the results achieved with both the conventional PIO algorithm and GA.

The Improvements in voltage levels before and after the restauration are depicted in Figure 4, visually showing the enhancement in node voltages throughout the network following the application of the proposed IPIO algorithm.

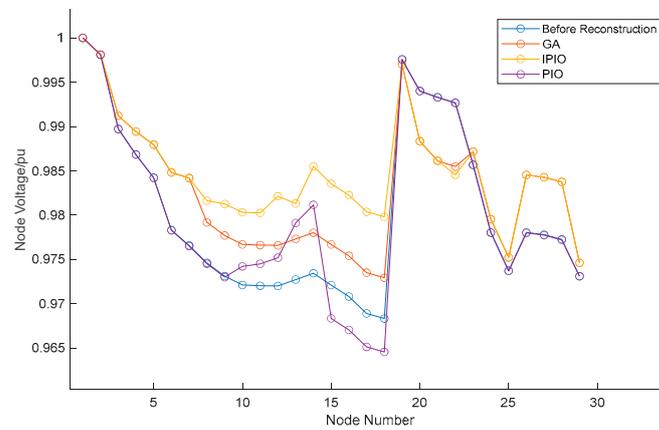


Figure 4. Node voltage before and after restauration (Fault on Branch 28).

### 5.2.2. Simultaneous Faults on Branches 25 and 28

The islanded sections remain unchanged, but both Branch 25 and Branch 28 experience faults in this scenario. Branch 25 has malfunctioned, which can be equivalent to disconnection, and simultaneously, a fault in Branch 28 occurs. The switch S29 is disconnected, leaving the nodes 25, 26, and 27 in the power outage state. To restore power, node 28 is reconnected to the main grid by engaging the tie switch S37. The fault recovery results are summarized in Table 5.

Table 5. Results of fault recovery and restauration (Faults on Branches 25 and 28).

Algorithm	Minimum Voltage/pu	Active Power Loss/kW	Disconnected Switch Number	Switching Operations Count
Before	0.9715	42.2289	S33, S34, S35, S36, S37	--
IPIO	0.9754	38.1056	S29, S36, S7, S11, S13	8
PIO	0.9677	45.0615	S29, S36, S15, S33, S35	4
GA	0.9731	38.6342	S29, S36, S7, S34, S35	4

It can be understood from Table 5 that when both Branch 25 and Branch 28 experience faults simultaneously, applying the proposed IPIO algorithm to solve the distribution network fault recovery problem leads to a reduction in active power losses by 4.1233 kW compared to the pre-restoration scenario. Additionally, the minimum node voltage in the network increases by 0.0039 pu. Overall, considering various factors, the proposed IPIO algorithm demonstrates significant improvement over both the conventional PIO and GA algorithms. The specific improvements in voltage after restauration are illustrated in Figure 5.

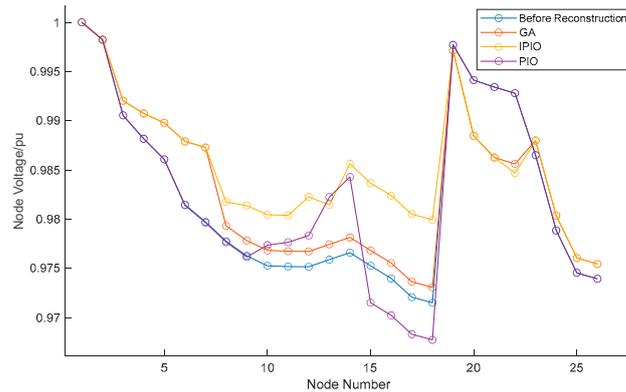


Figure 5. Node voltage before and after restoration (Faults on Branches 25 and 28).

### 5.3. Algorithm Performance Comparison

Firstly, in the conventional PIO algorithm, half of the individuals with lower fitness are eliminated in each iteration. Even if the initial population is very large, the population size decreases sharply after several iterations, which is not suitable for preserving the population diversity. The proposed IPIO algorithm introduces an elimination rate, with which only 10% of the poor individuals are eliminated in each iteration, effectively preserving the global optimal individuals. For example, when simulating a fault on branch 28 for recovery calculation, the remaining number of pigeon groups after each iteration should be recorded and a graph drawn. As shown in Figure 6, both algorithms (conventional PIO and the proposed case) start with a population of 300. The conventional PIO algorithm retains only two individuals after the 8th iteration, whereas the IPIO algorithm still has nine individuals after the 38th iteration.

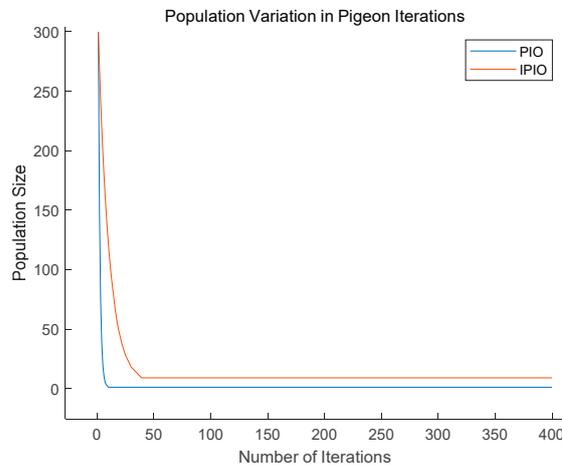


Figure 6. Population variation in pigeon iterations.

Furthermore, the conventional PIO algorithm is divided into two phases: the geomagnetic phase and the landmark phase. After adding the chaos strategy and the Cauchy perturbation redistribution strategy in the landmark phase, when a single fault or multiple faults occur or when simulating a fault in branch 28 for recovery calculation, the fitness values of each generation’s objective function should be calculated and plotted into a graph. It is evident from Figure 7 that both the conventional PIO and proposed IPIO have similar fitness values for the objective function in the first phase, both around 0.89. However, when the proposed IPIO enters the landmark phase, it can overcome the PIO’s tendency to get stuck in local optima, thereby finding individuals with a better fitness value of 0.77. During the recovery calculation process after simulating a fault in the distribution network,

this demonstrates that the proposed IPIO algorithm is suitable for distribution network restoration and can more accurately find the optimal strategy for network recovery.

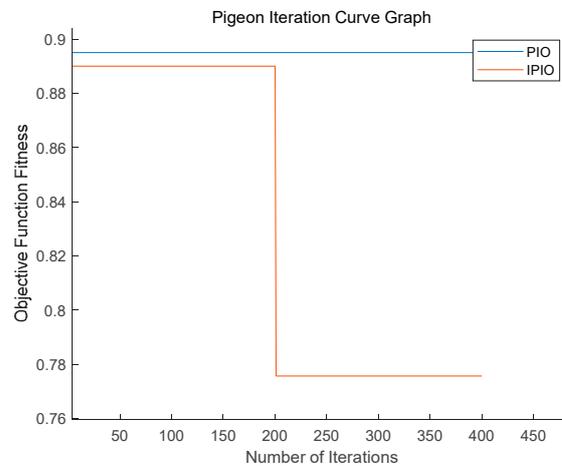


Figure 7. Pigeon iteration curve graph.

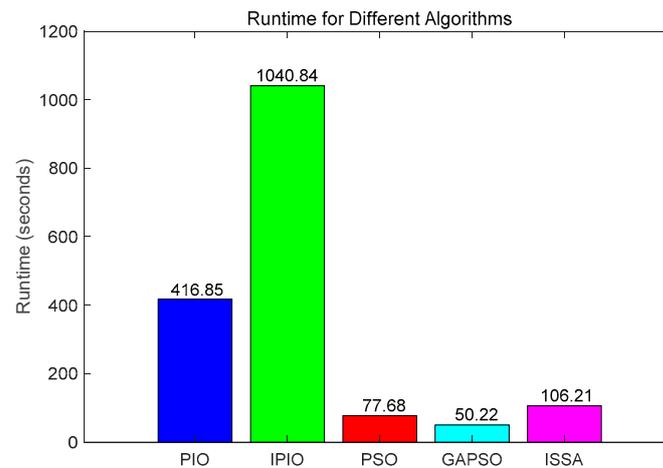
To test the performance of the IPIO, 10 repeated fault recovery experiments were conducted using the proposed algorithm, Particle Swarm Optimization (PSO), Genetic Particle Swarm Optimization (GAPSO), and Improved Sparrow Search Algorithm (ISSA) for the considered fault scenarios. The fault recovery results are illustrated in Table 6.

Table 6. Performance comparisons of algorithm.

Algorithm Name	Average Number of Iterations	Number of Times Trapped in Local Optima
IPIO	15.8	0
PSO	26.5	5
GAPSO	10	1
ISSA	98.2	5

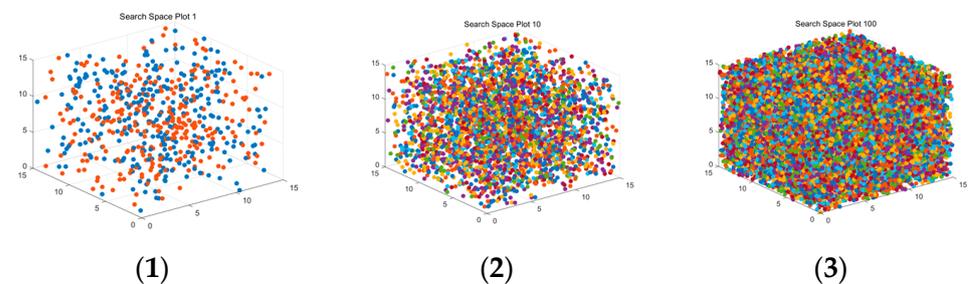
It can be observed from Table 6 that the ISSA algorithm had the worst performance. Although the PSO algorithm had fewer iterations, it was also prone to getting stuck in local optima and did not find the global minimum. In comparison, both the GAPSO algorithm and the proposed IPIO algorithm presented in this paper performed better. Although the IPIO algorithm had more iterations, it experienced fewer instances of getting stuck in local optima compared to the GAPSO algorithm and had a higher probability of achieving the optimal solution. Therefore, the proposed IPIO algorithm outperforms the other three algorithms in terms of topological optimization for distribution network recovery.

Figure 8 depicts a comparison of the running times of the considered five algorithms. The computer configuration used is AMD Ryzen 7 7840HS with Radeon 780M Graphics 3.80 GHz, with 16 GB of memory. From the graph, it can be observed that the conventional PIO algorithm requires a longer time due to its requirements for a higher number of iterations in both of its navigation units. Owing to its increased complexity, additional improvements, and reduced population complexity, the proposed IPIO algorithm has the longest runtime. Although the algorithm’s complexity and runtime have increased, the results indicate that the proposed IPIO algorithm effectively overcomes the issue of getting stuck in local optima, exhibits higher convergence, and can obtain superior solutions. Furthermore, when faced with different problems, it can adapt to more complex environments by adjusting parameters and improving its speeds.



**Figure 8.** Runtime for different algorithms.

Figure 9 depicts the search space of the IPIO algorithm. By incorporating a chaos and reverse strategy, the initial population positions generated by the IPIO are more diverse, covering a wider spatial range. The positions of each generation of population are recorded and plotted on a three-dimensional scatter plot for comparative analysis. Figure 9(1) illustrates the initial positions of pigeons generated by the IPIO algorithm in a three-dimensional space. Figure 9(2) displays the spatial distribution of pigeon positions after 10 iterations. As shown in Figure 9(3), after 100 iterations, it is evident that the IPIO algorithm has effectively explored most of the possible positions in the three-dimensional space. This demonstrates the algorithm's strong searching capability. With the improvements introduced in this paper, the proposed IPIO algorithm overcomes the drawback of being prone to local optima and is well-suited for distribution network fault recovery problems.



**Figure 9.** IPIO algorithm search space plot.

## 6. Conclusions

This paper focused on addressing the fault recovery problem in distribution networks with distributed energy sources. The fast convergence and high search efficiency of the PIO algorithm were enhanced and applied to find the global optimal solution for the nonlinear problem of distribution network recovery. Initially, a mathematical model for fault restoration in distribution networks with distributed energy sources was formulated, considering normalized objective functions and suitable constraints. Secondly, the Kruskal algorithm was used to implement a predefined islanding scheme, optimizing the distribution network topology based on various islanding constraints. Finally, an improved version of the conventional PIO algorithm was introduced in this paper. The proposed approach incorporated chaos and reverse strategies in generating the initial population and retained superior individuals through elite selection. As a result, the covered solution space was broadened. The algorithm incorporated a criterion to escape local optima and introduced Cauchy perturbation when trapped in local optima. Further, the elimination rate of the iteration operator was adjusted to preserve population diversity.

The proposed approach was validated using the IEEE 33-node distribution system through simulations. The results demonstrated improvements in node voltages and a reduction in active power losses compared to the pre-restoration network. Furthermore, when comparing the conventional PIO algorithm with the genetic algorithm, it was evident that the enhanced PIO algorithm exhibited superiority in enhancing the global optimal solution for distribution network fault recovery problems.

**Author Contributions:** Conceptualization, M.L. and J.W.; methodology, J.W.; software, M.L.; validation, M.L., J.W. and Q.Z.; formal analysis, H.Z.; investigation, M.L.; resources, M.L.; data curation, Q.Z.; writing—original draft preparation, M.L.; writing—review and editing, M.L.; visualization, M.L.; supervision, J.W. and Q.Z.; project administration, H.Z.; funding acquisition, J.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the open project of Key Laboratory in Xinjiang Uygur Autonomous Region of China, under Grant (2023D04071), the National Natural Science Foundation of China, under Grant (52167016), and a project supported by the Key Research and Development Project of Xinjiang Uygur Autonomous Region (2022B01020-3).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The anonymized data used in this study are available on request from the corresponding author.

**Conflicts of Interest:** Author Qiang Zhang was employed by the company State Grid Xinjiang Integrated Energy Service Co., Ltd. and author Hongjuan Zheng was employed by the company Nari Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## References

1. El-Gohary, A.; Al-Ruzaiza, A.S. Chaos and adaptive control in two prey, one predator system with nonlinear feedback. *Chaos Solitons Fractals* **2007**, *34*, 443–453. [\[CrossRef\]](#)
2. Liu, Y.; Chen, Z.; Yuan, Z. An Improved Particle Swarm Optimization and Its Application in Investment Operation. *Math. Pract. Theory* **2007**, *1*, 82–87.
3. Tizhoosh, H.R. In Opposition-Based Learning: A New Scheme for Machine Intelligence. In Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), Vienna, Austria, 28–30 November 2005; pp. 695–701.
4. Rudolph, G. Local convergence rates of simple evolutionary algorithms with Cauchy mutations. *IEEE Trans. Evol. Comput.* **1997**, *1*, 249–258. [\[CrossRef\]](#)
5. Stacey, A.; Jancic, M.; Grundy, I. Particle swarm optimization with mutation. In Proceedings of the 2003 Congress on Evolutionary Computation, CEC'03, Canberra, Australia, 8–12 December 2003; Volume 2, pp. 1425–1430.
6. Liu, Y.; Wu, Z.; Tu, Y.; Huang, Q.; Luo, W. A Survey on Distributed Generation and Its Networking Technology. *Power Syst. Technol.* **2008**, *15*, 71–76.
7. Qian, K.; Yuan, Y.; Cheng-ke, Z. Study on Impact of Distributed Generation on Distribution System Reliability. *Power Syst. Technol.* **2008**, *11*, 74–78.
8. Payne, J.; Gu, F.; Razeghi, G.; Brouwer, J.; Samuelsen, S. Dynamics of high penetration photovoltaic systems in distribution circuits with legacy voltage regulation devices. *Int. J. Electr. Power Energy Syst.* **2021**, *124*, 106388. [\[CrossRef\]](#)
9. Shen, X.; Shahidehpour, M.; Zhu, S.; Han, Y.; Zheng, J. Multi-Stage Planning of Active Distribution Networks Considering the Co-Optimization of Operation Strategies. *IEEE Trans. Smart Grid* **2018**, *9*, 1425–1433. [\[CrossRef\]](#)
10. Wang, B.; Zhu, H.; Xu, H.; Bao, Y.; Di, H. Distribution Network Reconfiguration Based on NoisyNet Deep Q-Learning Network. *IEEE Access* **2021**, *9*, 90358–90365. [\[CrossRef\]](#)
11. Fan, L.; Si, W.; Xuan, Y.; Sun, Z.; Zhao, J.; Xu, B.; Gu, Q. Multi-Objective Optimal Configuration of Multiple Switchgear Considering Distribution Network Fault Reconfiguration. *IEEE Access* **2021**, *9*, 69905–69912. [\[CrossRef\]](#)
12. Shukla, J.; Panigrahi, B.K.; Ray, P.K. Stochastic reconfiguration of distribution system considering stability, correlated loads and renewable energy based DGs with varying penetration. *Sustain. Energy Grids Netw.* **2020**, *23*, 100366. [\[CrossRef\]](#)
13. Senjyu, T.; Miyazato, Y.; Yona, A.; Urasaki, N.; Funabashi, T. Optimal Distribution Voltage Control and Coordination with Distributed Generation. *IEEE Trans. Power Deliv.* **2008**, *23*, 1236–1242. [\[CrossRef\]](#)
14. Tang, Y.; Wu, Z.; Gu, W.; Yu, P.; Du, J.; Luo, X. Research on Active Distribution Network Fault Recovery Strategy Based on Unified Model Considering Reconstruction and Island Partition. *Power Syst. Technol.* **2020**, *44*, 2731–2740.
15. Huang, J.; Zhang, Y.; Li, Q. Application of Ant Colony System in Distribution Reconfiguration. *Proc. CSU-EPSA* **2007**, *19*, 59–64.

16. Yu, J.; Zhang, F. Distribution Network Reconfiguration Based on Improved Immune Genetic Algorithm. *Power Syst. Technol.* **2009**, *33*, 100–105.
17. Li, Z.; Chen, X.; Yu, K.; Liu, H.; Zhao, B. Hybrid Particle Swarm Optimization for Distribution Network Reconfiguration. *Proc. CSEE* **2008**, *28*, 35–41.
18. Lei, S.; Wang, S.; Hu, X.; Zhou, Q.; Lin, M. Chaotic Optimization Integrated with Artificial Immune Algorithm for Distribution Service Restoration after Faults. *High Volt. Eng.* **2009**, *35*, 1492–1496.
19. Wang, F.; Chen, C.; Li, C.; Cao, Y.; Li, Y.; Zhou, B.; Dong, X. A Multi-Stage Restoration Method for Medium-Voltage Distribution System with DGs. *IEEE Trans. Smart Grid* **2017**, *8*, 2627–2636. [[CrossRef](#)]
20. Cao, F.; Zhang, Y.; Li, S. Dynamic Reconfiguration of Active Distribution Network Based on Similarity and Adaptability of Network Structure. *Autom. Electr. Power Syst.* **2019**, *43*, 78–85.
21. Duan, H.; Qiao, P. Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning. *Int. J. Intell. Comput. Cybern.* **2014**, *7*, 24–37. [[CrossRef](#)]
22. Huo, M.; Wei, C.; Yu, Y.; Zhao, J. Clustering optimization algorithm for large-scale unmanned aerial vehicle based on intelligent behavior of pigeons. *Sci. Sin. (Technol.)* **2020**, *50*, 475–482.
23. Pham, T.T.H.; Besanger, Y.; Hadsaid, N. New Challenges in Power System Restoration with Large Scale of Dispersed Generation Insertion. *IEEE Trans. Power Syst.* **2009**, *24*, 398–406. [[CrossRef](#)]
24. Zheng, H. *Research on Fault Recovery Reconstruction of Distribution Network Based on GA-BFGS Algorithm*; China University of Mining and Technology: Xuzhou, China, 2020.
25. Chen, Q.; Wang, W.; Wang, H.; Wu, J.; Li, X.; Lan, J. A Social Beetle Swarm Algorithm Based on Grey Target Decision-Making for a Multiobjective Distribution Network Reconfiguration Considering Partition of Time Intervals. *IEEE Access* **2020**, *8*, 204987–205013. [[CrossRef](#)]
26. Liu, Z.; Bao, Q.; Sun, C.; Wu, X. Islanding algorithm of distribution system with distributed generations based on improved Kruskal algorithm. *Trans. China Electrotech. Soc.* **2013**, *28*, 164–171.
27. Hao, Z.; Ji-hong, S.; Tie-nan, Z.; Yang, L. An Improved Chaotic Particle Swarm Optimization and Its Application in Investment. In Proceedings of the International Symposium on Computational Intelligence and Design, Washington, DC, USA, 17–18 October 2008; pp. 124–128.
28. Tian, S.; Liu, L.; Wei, S.; Fu, Y.; Yang, M.; Liu, S. Dynamic reconfiguration of a distribution network based on an improved grey wolf optimization algorithm. *Power Syst. Prot. Control* **2021**, *49*, 1–11.
29. Xu, Z.; Pan, J.; Fan, S.; Song, X.; Zhai, S.; Gong, M. A distribution network reconfiguration method with distributed generation based on improved firefly algorithm. *Power Syst. Prot. Control* **2018**, *46*, 26–32.
30. Chen, J.; Xiao, Z. Research on adaptive genetic algorithm based on multi-population elite selection strategy. In Proceedings of the IEEE International Conference on Computational Intelligence and Applications (ICCI), Beijing, China, 8–11 September 2017; pp. 108–112.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.