Research Article

Improved Particle Swarm Optimization Algorithm in Power System Network Reconfiguration

Yanmin Wu1,2 and Qipeng Song3

1College of Electric Engineering, Naval University of Engineering, Wuhan 430033, Hubei, China
2School of Building Environment Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, Henan, China
3China Electric Power Research Institute, Beijing 100192, China

Correspondence should be addressed to Yanmin Wu; 2006121@zzuli.edu.cn

Received 8 January 2021; Revised 19 February 2021; Accepted 26 February 2021; Published 11 March 2021

Copyright © 2021 Yanmin Wu and Qipeng Song. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the rapid development of the social economy, the rapid development of all social circles places higher demands on the electricity industry. As a fundamental industry supporting the salvation of the national economy, society, and human life, the electricity industry will face a significant improvement and the restructuring of the network as an important part of the power system should also be optimised. This paper first introduces the development history of swarm intelligence algorithm and related research work at home and abroad. Secondly, it puts forward the importance of particle swarm optimization algorithm for power system network reconfiguration and expounds the basic principle, essential characteristics, and basic model of the particle swarm optimization algorithm. This paper completes the work of improving PSO through the common improved methods of PSO and the introduction of mutation operation and tent mapping. In the experimental simulation part, the improved particle swarm optimization algorithm is used to simulate the 10-machine 39-bus simulation system in IEEE, and the experimental data are compared with the chaos genetic algorithm and discrete particle swarm optimization algorithm. Through the experimental data, we can know that the improved particle swarm optimization algorithm has the least number of actions in switching times, only 4 times, and the chaos genetic algorithm and discrete particle swarm optimization algorithm are 5 times; compared with the other two algorithms, the improved particle swarm optimization algorithm has the fastest convergence speed and the highest convergence accuracy. The improved particle swarm optimization algorithm proposed in this paper provides an excellent solution for power system network reconfiguration and has important research significance for power system subsequent optimization and particle swarm optimization algorithm improvement.

1. Introduction

Electricity is the main energy base of a country, related to saving the national economy. The development of modern electricity has entered a period of multiple services, which is linked to global resources, environmental protection, and sustainable development. At present, with the development of economy and the improvement of people’s living standards, the requirements of consumers for electric energy are constantly improving. The requirements of “safety, reliability, economy, high quality, and environmental protection” in the power supply are also constantly improving, and the power system is developing in the direction of automation, optimization, adaptation, intelligence, coordination, and regionalization. For the time being, China has entered the stage of integrated electricity generation. In energy production, rational planning of production, transmission, distribution, and transformation not only improves the reliability of the electricity network, but it also improves the economy of the electricity network, saving human and economic resources.

However, the power system has nonlinear, multiconstrained, unconventional, and high-dimensional optimization problems. However, the traditional heuristic algorithm is not a strict optimization method and cannot deal with the interaction between various objective functions. The
mathematical optimization method is very difficult to calculate the optimization problem, which is easy to cause dimension disaster. The swarm intelligence algorithm includes genetic algorithm, simulated annealing algorithm, tabu algorithm, ant colony algorithm, fish swarm algorithm, and particle swarm optimization. Among them, the particle swarm optimization algorithm has the advantages of fast convergence speed, multiparticle parallel processing, and easy application, which can solve the optimization problem of the power system.

The concept of swarm intelligence was first proposed by American scholars Hackwood and Beni in the molecular automation system. Inspired by the collective behavior of natural groups, the swarm intelligence algorithm has been widely used in life [1]. Cheng is based on the combination of the swarm intelligence algorithm and data mining, so that big data problems can be better understood and analyzed [2]. American scholars Eberhart and Kennedy studied the foraging behavior of birds and proposed particle swarm optimization (PSO). The main theoretical basis of the algorithm is artificial life and evolutionary information theory, and its basic idea is to find the best solution through the cooperation between individuals. Since the particle swarm optimization was proposed, in order to improve the performance of the algorithm, a large number of researchers at home and abroad have improved it. These improved algorithms mainly have the following ideas: improving the inertia weight of its own parameters, improving its own parameter learning factor, and combining with other algorithms. Mahi introduced the 3-opt algorithm into the particle swarm optimization algorithm to optimize the performance of the algorithm to improve the defects of the local solution of the particle swarm optimization algorithm [3]. Gong proposed a new particle swarm optimization algorithm called the genetic particle swarm optimization algorithm, which uses genetic evolution technology to cultivate the particle swarm optimization algorithm [4].

The optimization algorithm adopted in this article is based on the classical particle speed optimization algorithm, which accelerates the rate of convergence and improves the accuracy of convergence. Calculation results have little chance, good stability, and a strong global search capability. This article is mainly concerned with improving the PSO learning method and the optimization algorithm. The model of analysis of convergence theory and the improved particle optimization algorithm shall be used to study the reformulation of the power grid. The main purpose of this work is to propose a more appropriate and valuable algorithm for the restructuring of the power grid.

2. Basic Model and Improved Method of Particle Swarm Optimization

2.1. Basic Principle of Particle Swarm Optimization Algorithm. Since Kennedy and Eberhart first proposed the particle cluster optimization algorithm, more and more attention has been drawn to and research by domestic and foreign researchers and has proposed various improved versions and applications from one angle [5]. Particle swarm optimization is an evolutionary algorithm based on population, and its ideological source is the theory of artificial life and evolutionary computing [6]. According to Reynolds’ research on bird flight, a single bird only needs to track a limited number of birds around it, but the final result is that the whole flock seems to fly orderly under an invisible control. In other words, the complex and elegant global behavior is generated by the simplest rule interaction, which is the essence of swarm intelligence. Particle swarm optimization is derived from the study of foraging behavior of birds: a group of birds are looking for food randomly. If there is only one piece of food in this area, then the simplest and most effective search strategy is to find food in the nearest place to the food. Particle swarm optimization is inspired from this mode and further used to solve optimization problems [7, 8]. In addition, people usually make decisions based on their own and others’ experience, which is also in line with the basic idea of particle swarm optimization: when particle swarm optimization is used to solve optimization problems, the solution of the problem corresponds to the position of a bird (particle) in the search space. The particle not only has its own position and velocity but also has a suitable value determined by the objective function. Each particle remembers and follows the current optimal particle and searches the solution space: each search (iteration) contains some random factors, but these factors are not completely random. If a better solution is found, the next good solution can be found on this basis. Specifically, the particle swarm optimization starts with a random initial particle. In each iteration, the particle updates its position by tracking two “extreme points”: one is the best solution found by the particle itself (i.e., individual extreme point, pbest) and the other is the extreme point (gbest) of the whole particle swarm or the extreme point of the whole neighborhood (lbest). On the basis of the original inertia, particles adjust the flight direction and speed according to these two extrema to maintain the overall optimum [9, 10].

2.2. Essence and Characteristics of Particle Swarm Optimization.

(1) Particle swarm optimization is a new swarm intelligence optimization technology, which is the simulation of group society. After much research and practice by the PSO, the substance of the PSO can be summed up as “random” and “leadership.” Random is the blood of particle optimization. Particulate optimization is based on the Monte Carlo algorithm of the previous generation, which fully inherits the characteristics of random number resolution. Leadership is the evolution of the particle swarm optimization algorithm. It simulates the phenomenon of mutual cooperation in a group society. Individuals have the ability to learn, and the group has the ability to cooperate with each other. According to the guidance of leader particles, the whole population will converge to the global optimal solution [11, 12]. Therefore, the advantages and disadvantages of PSO are easy to understand. Due to the randomness of
2.3. Basic Model of Algorithm. Particle optimization is a smart algorithm based on repetitive function. His basic idea is to randomly prepare a particle group with memory but without volume and mass. Each particle represents a feasible solution to the optimization problem and determines a moderate value through the measurement function. Good and bad particle count is judged by the appropriate value [15]. Each particle has the ability of memory and can adjust its own motion trajectory according to its current position, information sharing between peers, and the best position experienced in memory. After iteration, the particle keeps approaching the best position and finally reaches the optimal position [16].

The mathematical description of the basic particle swarm optimization algorithm is as follows.

Assuming that the population size of a particle is n, the position information of the \(i\)th particle \((i = 1, 2, \ldots, n)\) in \(d\)-dimensional space is represented by

\[
x_i = (x_{i1}, x_{i2}, \ldots, x_{id}, \ldots, x_{id}),
\]

and the velocity of the \(i\)th particle is expressed by \(v_i\)

\[
v_i = (v_{i1}, v_{i2}, \ldots, v_{id}, \ldots, v_{id}).
\]

Thus, at time \(t + 1\) \((t = 1, 2, \ldots, T_{\text{max}})\), the flight velocity \(v_{id}\) of the \(i\)th particle in the \(d\)-dimensional \((d = 1, 2, \ldots, d)\) subspace is adjusted according to the following formula:

\[
v_{id}(t + 1) = v_{id}(t) + c_1r_1(t)(p_{id}(t) - x_{id}(t)) + c_2r_2(t)(P_{\text{gd}} - x_{id}(t)).
\]

In formula (3), \(p_{id}\) represents the historical optimal position of the current particle, \(P_{\text{gd}}\) represents the global historical optimal position of the particle, and \(c_1\) and \(c_2\) are nonnegative acceleration constants, also known as convergence factors. The convergence factor \(c_1\) is called cognitive constant, which represents the characteristics of particles learning from their own optimal state. \(r_1\) and \(r_2\) are a random number distributed in the interval [0,1]. It can be seen from the formula that the first the flying speed of a particle is mainly determined by the following three parts: the first part is the speed of the particle at time \(t\), which represents the particle’s trust in the flight speed at time \(t\); the second part is the cognitive part of the particle itself, which represents the particle’s thinking about its position, thinking about the position before the particle itself, so as to determine the next step. The third part is the “social” part of particles, which represents the information exchange and cooperation between particles and their peers. In the process of searching, particles synthesize their own previous flight experience and the experience of their companions and finally determine their own flight speed according to formula (3).

Weight of inertia \(W\): the global search capability of the main population is controlled by global search speed and inertia. If \(W\) gets a fixed higher value, the algorithm convergence speed will be too slow and the final accuracy of the solution is also very low. This is because the position of particles changes greatly each time, and it is very easy to miss the optimal value region, which leads to the algorithm is difficult to determine the existence area of the optimal value and does not have the local fine search ability; if the fixed small value of \(W\) is taken, the algorithm can detect the local region. Because the search step size is small and the particle position changes little after each update, it is difficult to traverse the whole search space, so it is difficult to locate the region where the optimal value exists. Therefore, the algorithm is prone to premature convergence and premature in this case. In addition, it is found that the inertia weight setting value of PSO is too high. It is the best to decrease between 0.9 and 0.4, because the population has a large inertia weight at the beginning of evolution and can traverse a wider range, which is helpful to find the region where the optimal solution may exist. In the later stage of population evolution, the inertia weight is small, and the algorithm has strong local search ability, which is helpful to find the best solution in the field where the optimal solution exists [17, 18].

Convergence factors \(c_1\) and \(c_2\): these two parameters represent the acceleration weights of particles approaching individual and global extremum, respectively. The size of \(c_1\) determines the cognitive ability of particles, that is, the ability of particles to learn from themselves. In extreme cases, when \(c_1 = 0\), particles have no cognitive ability and are only affected by the “social part.” At this time, the population has the maximum ability to traverse the whole feasible solution space, and the convergence speed is fast. However, when solving the complex multipeak problems, it is very easy to find the local optimal problem. The size of \(c_2\) determines the social information sharing ability of particles, that is, the ability of particles to advance toward the current global optimal value. In extreme cases, \(c_2 = 0\), particles in the population will not share information. The algorithm only has the “cognitive” model. Since there is no information interaction between particles, it is very unlikely that the algorithm can get the optimal solution.

2.4. Algorithm Flow. Particle swarm optimization can be summarized as the following steps:

1. The position and velocity of all particles (population size \(n\)) and the basic parameters of the algorithm are initialized.

PSO, it has advantages in solving nonconvex, discontinuous, high-dimensional, nonlinear, and non-differentiable optimization problems. Similarly, due to the randomness, its calculation results may be randomness. If the leadership is enhanced, the randomness of the calculation results of the algorithm will be reduced; due to the strong leadership of the PSO, the PSO may easily fall into If the randomness is increased, the algorithm may jump out of the local optimum and find the global optimal [13, 14]. The essence of particle swarm optimization “random” and “leader” have the characteristics of "spear" and "shield," which need a kind of thought of trade-off to balance.
(2) Taking the initial position of each particle as individual extremum, the initial fitness value of each particle in the population is calculated, and the optimal position of the current population is obtained.

(3) Update the speed and position of each particle.

(4) Compare the current fitness value with the historical optimal value. If the current fitness value is superior to the historical optimal value, the historical optimal value of the particle is set to the current fitness value, and the particle position is updated.

(5) If the current fitness value is better than the historical population optimal value, the historical population optimal value is set as the current fitness value to update the global optimal solution of the population.

(6) Check the end conditions. If the optimization results meet the conditions for the iteration stop, the optimal solution is obtained and the iteration is terminated. Otherwise, turn back to step 4 and continue the optimization until the iteration stop condition is met.

2.5. Several Common Improved Particle Swarm Optimization Algorithms

2.5.1. Particle Swarm Optimization Model with Inertia Weight. Shi and Eberhart add the inertia weight parameter to the speed update formula of the basic particle swarm optimization algorithm, which is multiplied by the particle velocity at \( t \) time. This method makes the algorithm achieve the effective balance between global search and local search by adjusting the value of inertia weight and improves the search ability of the algorithm. The main purpose of introducing inertia weight parameters is to better balance the global detection and local search capabilities of particle swarm optimization [19]. The results show that when \( w \) is set larger, it is beneficial to global breadth search; conversely, it is beneficial to local depth search. This improvement is achieved by adding an inertia coefficient \( w \) to the speed update formula; that is, the new speed update formula is as follows:

\[
v_{id}(t) = \omega v_{id}(t-1) + c_1 \text{rand} \{ p_{best_id}(t-1) - p_{id}(t-1) \} + c_2 \text{rand} \{ g_{best}(t-1) - p_{id}(t-1) \},
\]

where \( \omega \) is the inertia weight.

2.5.2. Adding Convergence Factor. Clerc adds a convergence factor \( \chi \) to the speed update formula of the basic particle swarm optimization algorithm to ensure the convergence of the algorithm. This method is actually \( v_i \) reselection of parameters \( \omega, c_1, \) and \( c_2 \). By properly selecting these controllable parameters, the value of \( a \) does not have to be limited in \([-v_{max}, v_{max}]\). In this case, the corresponding algorithm speed update formula can be expressed as follows:

\[
v_{id}(t) = \chi \left[ v_{id}(t-1) + c_1 \text{rand} \{ p_{best_id}(t-1) - p_{id}(t-1) \} + c_2 \text{rand} \{ g_{best}(t-1) - p_{id}(t-1) \} \right].
\]

In the experiment, \( \chi = \frac{2}{1 - \phi - \sqrt{\phi^2 - 4 \phi}} \), \( \phi = c_1 + c_2 > 4 \), \( \phi \) is usually taken as 4.1 (at this time, \( c_1 = c_2 = 2.05 \), i.e., \( \chi = 0.729 \)), which is equivalent to \( w = 0.729, c_1 = c_2 = 1.1494 \) in formula (5), which is called the particle swarm optimization algorithm with convergence factor. Experimental data show that the particle swarm optimization with convergence factor effectively overcomes the shortage of linear decreasing weight and has its unique effect compared with the inertia weight of other formats. Therefore, this technology has been paid attention to in the design of the particle swarm optimization algorithm [20].

2.5.3. Differentiation Algorithm. The basic idea of the differential evolution algorithm is to take the differential carrier of two randomly selected individuals in the population as a source of random change of the third person, i.e., to create the mutant person, and then the new person is created by the crossover function between the mutant person and the target person. If the value and the suitability of the new person are better than those of the target person, the target person will be informed as a new person to enter the next generation. Otherwise, the target person will be preserved in the next generation [21,22].

The implementation process of differentiation algorithm is as follows:

(1) Generating initial population

\( M \) individuals are randomly generated in \( n \)-dimensional space:

\[
x_{ij}(0) = x_{ij}^L + \text{rand}(0, 1)(x_{ij}^U - x_{ij}^L),
\]

where \( \text{rand}(0, 1) \) is a random number between \([0, 1]\).

(2) Mutation operation

The key step of differential evolution is mutation operation, which randomly selects three individuals \( x_{p1}, x_{p2}, x_{p3} \) and \( p1 \neq p2 \neq p3 \neq i \) from the feasible solutions:

\[
h_{ij}(g) = x_{p1j} + F(x_{p2j} - x_{p3j}).
\]

(3) Cross operation

The main purpose of cross operation is to increase the diversity of feasible solutions:

\[
v_{ij}(g + 1) = \begin{cases} h_{ij}(g), & \text{if } r \text{ and } (0, 1) \leq CR \text{ or } j = \text{rand}(1, n), \\ x_{ij}(g), & \text{if } r \text{ and } (0, 1) > CR \text{ or } j \neq \text{rand}(1, n), \end{cases}
\]

where \( CR \) is the crossover probability, \( CR \in [0, 1] \), and \( \text{rand}(1, n) \) is a random integer between \([1, n]\).
In order to update the template individuals, vector \( v_{ij}(g+1) \) and vector \( x_i(g) \) were compared:

\[
x_i(g+1) = \begin{cases} 
  v_i(g+1), & \text{if } f(v_i(g+1)) < f(x_i(g)), \\
  x_i(g), & \text{if } f(v_i(g+1)) \geq f(x_i(g)).
\end{cases}
\]

(9)

The mutation operation is repeated until the required convergence precision is reached or the maximum evolution algebra is reached.

### 2.6. Discrete Mode of Particle Swarm Optimization.

Power system network reconfiguration is a discrete multiobjective combinatorial optimization problem. The final result of the reconfiguration is a group of switching action combinations. For the power grid model used in this paper, the reconstruction result can be regarded as a 20-dimensional vector. Therefore, in order to make the particle swarm optimization algorithm solve the problem of this topic, it must be discretized chemical treatment.

At first, particle swarm optimization (PSO) can only be directly used to solve the continuous optimization problem. The discrete binary particle swarm optimization (dbpso) mainly adjusts the speed update mode of particle swarm optimization and introduces the probability function:

\[
\begin{align*}
  v_{id}^{k+1} &= v_{id}^k + c_1 r_1(i_{id} - x_{id}) + c_2 r_2(P_{id} - x_{id}), \\
  x_{id}^{k+1} &= \begin{cases} 
    1, & \text{rand} < S(v_{id}^{k+1}), \\
    0, & \text{rand} \geq S(v_{id}^{k+1}).
  \end{cases}
\end{align*}
\]

(10)

In the formula, \( \text{Rand} \) is the random number of \([0, 1]\); the function \( S \) represents the probability function in the interval \([0, 1]\) determined by the velocity \( a \). The closer the \( S \) is to 1, the more likely it is to take 1; otherwise, the more likely it is to take 0. Generally, the fuzzy function sigmoid is used as the probability function. The sigmoid function is as follows:

\[
\text{Sigmoid}(v_{id}^{k+1}) = \begin{cases} 
  0, & v_{id}^{k+1} < V_{\text{min}}, \\
  1 + \exp(-v_{id}^{k+1}), & V_{\text{min}} \leq v_{id}^{k+1} < V_{\text{max}}, \\
  0.99, & v_{id}^{k+1} > V_{\text{max}}.
\end{cases}
\]

(11)

### 3. Improved Particle Swarm Optimization Algorithm

In order to make the improved particle swarm optimization algorithm more suitable for power system network reconfiguration, this paper introduces a variety of algorithms based on the abovementioned improved particle swarm optimization algorithm and uses the self-made parameters of the experimental simulation system to verify the performance of the improved particle swarm optimization algorithm.

#### 3.1. Definition of Population Diversity.

In general, the more concentrated the particles are in the search space, the lower the population diversity. The more dispersed the particle distribution, the higher the diversity of the population [23]. This paper adopts the definition of diversity in the following formula:

\[
\text{diversity}(S) = \frac{1}{|S||L|} \sum_{i=1}^{|S|} \sum_{j=1}^{|L|} \left( p_{ij} - \overline{p}_j \right)^2.
\]

(12)

In the formula, \( p_{ij} \) is the \( j \)-th variable of the \( i \)-th particle, \( \overline{p}_j \) is the average value of the \( j \)-th variable of all particles, \(|S|\) is the population size, \(|L|\) is the longest radius of the search space, and \( dim \) is the dimension of the solution space. The diversity \( (s) \) value reflects the average distance between all particles in the population and the center of the search space and reflects the density of particles. The smaller the diversity \( (s) \), the more concentrated the particles in the population, and the larger the diversity \( (s) \), the more dispersed the particles in the population.

#### 3.2. Introducing Mutation Operation.

The rapid decrease of population diversity will lead to premature convergence of PSO, which is the biggest defect of PSO. Therefore, the mutation operation is introduced into the algorithm to adjust the population diversity attenuation process of particle swarm optimization: in this paper, \( p_h \) mutation operator similar to the genetic algorithm is introduced in the mutation operation of the improved particle swarm optimization algorithm. The specific operation is as follows: set the mutation rate \( a \) and the mutation random quantity \( \rho_j \) on the particle position component, where \( j = 1, 2, \ldots, \dim \) and \( \rho_j \) is the interval \([0, 1]\). For the position component in the \( j \)-th dimension of a particle, if \( \rho_j < \rho_{th} \), the position component \( \rho_j \) of a particle in this dimension will be replaced by the position component of its adjacent position particle in the memory library, and it will change to the particle dimension component with higher fitness value at the approximate rate and change to the particle dimension position component with smaller fitness value with a smaller probability [24, 25].

#### 3.3. Initialization of Improved Particle Swarm Optimization Algorithm Based on Tent Mapping.

Speed is an important objective of the restructuring of the power grid. A better initialization formula can improve the speed of convergence of the algorithm to the best solution. The traditional form of random initialization does not favour convergence of the algorithm, and the likelihood of finding the best solution is also greatly reduced.

At present, most experts use chaos rules based on logistic map to initialize population. Chaotic motion has the characteristics of regularity, ergodicity, and randomness. Chaos initialization is proposed based on this characteristic. Chaos initialization can improve the quality of the initial solution and enhance the diversity of the initial population, which is conducive to the population traversing the whole feasible solution region in the early evolution stage.
However, different chaotic maps have different effects on the chaotic optimization process. The ergodic uniformity and search efficiency of tent map are better than the logistic map. Therefore, this paper considers using tent map to initialize population particles. The iterative formula of the tent chaotic mapping operator is as follows:

$$r_{k+1} = \begin{cases} 2r_k, & r_k \in [0, 0.5], \\ 2(1 - x_k), & r_k \in (0.5, 1], \end{cases}$$

(13)

where $k = 12, 3, m_o$. Let the initial value of $r_{k+1}$ take the random number in the interval $[0, 1]$. Through the iterative calculation of the above formula, the random number between $M$ can be obtained. Using the above model to generate chaotic variables, map them to the interval of decision variables $[x_{\min}, x_{\max}]$ and get the initial particle swarm optimization population.

3.4. Specific Steps of Power System Network Reconfiguration Simulation Based on Improved Particle Swarm Optimization Algorithm. The use of the improved particle swarm optimization algorithm to reconfigure the power grid is usually divided into the following steps, and the specific process is shown in Figure 1:

1. Code the primary and secondary load with 0, 1, and 2 codes, 0 represents that the standby path switch is open, 1 represents that the normal path switch is closed, and 2 represents the standby path switch is closed; for the third-level load, 0 is for the switch is open, and 1 is for the switch to remain closed.
2. The initial population size, initial mutation rate $\rho_0$, and initial inertia weight $W$ are set.
3. The fitness value of each particle is calculated, and the current position information of the particle is recorded in the memory database to update the individual history optimal and population global optimal.
4. Update the position and velocity of population particles.
5. The diversity value of the current population was calculated.
6. Update inertia weight $W$ and variation rate $\rho_0$.
7. Mutate the particles.
8. Check whether the maximum number of iterations is reached. If the maximum number of iterations is reached, the best fitness value and its location information are output. Otherwise, return to step 3 to continue iteration.

4. Network Reconfiguration Simulation Experiment Based on Improved Particle Swarm Optimization Algorithm

4.1. Experimental Simulation of Improved Particle Swarm Optimization Algorithm. In this paper, the IEEE 10-machine 39-bus simulation system is used to carry out the simulation experiment, and the existing fault occurs in 22 and 32 nodes. The simulation experiment parameters are set as follows: population size $M = 50$, population dimension $\dim = 10$, maximum iteration times $t = 50$, acceleration factor $c_1 = c_2 = 2$, initial inertia weight $W = 0.8$, and initial value of variation rate is 0.1.

It can be seen from Table 1 and Figure 2 that the number of switches decreases with the iteration of the algorithm, from the first generation of 10 times of global optimal solution switching times to the fifth generation of four times, and it remains stable in the subsequent iterations. Table 2 and Figure 2 reflect the change of the global optimal adaptive speed with the algorithm iteration. It can be seen that the
optimal fitness value reaches the maximum when the algorithm is iterated to the sixth generation, and the drag racing is stable. The last graph in Figure 2 shows the optimal particle coding after 50 iterations. All the final schemes of power network reconfiguration are that branches 14 and 19 are unloaded and branches 15 and 20 use standby power supply. This method can solve the fault problem safely, effectively, and quickly, so the improved particle swarm optimization algorithm used in this paper is feasible.

4.2. Comparison of Improved Particle Swarm Optimization Algorithm with Chaos Genetic Algorithm and Discrete Particle Swarm Optimization Algorithm.

In order to prove the performance of the improved particle swarm optimization algorithm, the chaos genetic algorithm and discrete particle swarm optimization algorithm are used to simulate the same fault, and the experimental results of the two algorithms are compared with the experimental results of the improved particle swarm optimization algorithm.

In this experiment, the New England 10-machine 39-bus simulation system is still used in the experiment. The fault of the experiment is set as 6 and 21 nodes, and the three algorithms are iterated 40 times each.

It can be seen from Table 3 and Figure 3 that the maximum switching times of the global optimal solution of the chaos genetic algorithm are 12, the minimum switching times are 5, and the minimum number of switching times is 28; the maximum switching times of the discrete particle swarm optimization algorithm are 20, the minimum switching times are 5, and the minimum number of switching times is only 22; the maximum switching times of the improved particle swarm optimization algorithm are 10, the minimum number of switching times is only 4 among the three algorithms, and the minimum number of switching times is 38 times in the three algorithms.

Table 4 and Figure 4 record the global optimal fitness values of the three algorithms with the iteration of the algorithm. The minimum value of the global optimal fitness of the chaos genetic algorithm is 5.49, the maximum value is 5.65, and the maximum number of times is 26; the minimum value of discrete particle swarm optimization algorithm is 5.27, the maximum value is 5.64, and the maximum number of times is 21; the improved particle swarm optimization algorithm is the best among the three algorithms, with the maximum value of 5.72, the minimum value of 5.53, and the maximum occurrence times of up to 38 times.

Figure 5 shows the optimal particle code obtained by the three algorithms through iteration. From the optimal particle code of the three algorithms, we can get the corresponding reconstruction scheme: the reconstruction scheme of the chaotic genetic algorithm and discrete particle swarm optimization algorithm is the same, branch 5 and 9 are
restored by the backup path, and the branches 4, 13, and 17 are unloaded. The improved particle swarm optimization algorithm is that the power supply of branch 5 and 9 is restored by the standby path, and branches 13 and 17 are unloaded.

By sorting out the results, we can know that in terms of switching operation times, the improved particle swarm optimization algorithm has the least number of actions, only 4 times, and the other two algorithms are 5 times; in terms of the earliest convergence algebra, the improved particle

---

### Table 3: Comparison of switching times.

| Optimal particle switching times               | Maximum | Minimum | Minimum number of times |
|-----------------------------------------------|---------|---------|-------------------------|
| CGA’s optimal particles                      | 12      | 5       | 28                      |
| DPSO’s optimal particles                     | 20      | 5       | 22                      |
| IPSO’s optimal particles                     | 10      | 4       | 38                      |

### Figure 3: Comparison of switching times.

---

### Table 4: Fitness comparison of optimal particles.

| Fitness of optimal particles                  | Maximum | Minimum | Maximum number of times |
|----------------------------------------------|---------|---------|-------------------------|
| CGA’s optimal particles                      | 5.65    | 5.49    | 26                      |
| DPSO’s optimal particles                     | 5.64    | 5.27    | 21                      |
| IPSO’s optimal particles                     | 5.72    | 5.53    | 38                      |

### Figure 4: Fitness comparison of optimal particles.
swarm optimization algorithm is the first to converge to the global optimum, and the chaos inheritance algorithm and discrete particle swarm optimization algorithm need a lot of iterations to achieve the global optimal; in the restorability, the improved particle swarm optimization algorithm can achieve the global optimization. In the aspect of energy, it is obvious that the restoration scheme provided by the improved particle swarm optimization algorithm can meet the requirements of maximum recovery of load power supply, and the convergence accuracy is the highest.

Through the above comparison, it can be seen that the improved particle swarm optimization algorithm has a certain degree of improvement in the number of iterations and the solution accuracy. The recovery scheme obtained is better than other algorithms, the switching operation cost is also low, and the convergence performance is also improved, which can ensure that a better recovery scheme can be finally obtained.

5. Conclusions

Under the background of the continuous expansion of the scale and complexity of the power system, how to ensure the realization of the operation control objectives of the power system and how to construct and solve various optimization problems in the power system under the constraints of many complex and uncertain conditions are always the important issues faced by power engineers. Power system network reconfiguration is an important part of modern power system intelligent management and an important guarantee for stable and reliable operation of power grid. Once a large-scale blackout occurs, if there is no predetermined recovery plan, it will cause great economic losses. Therefore, it is very necessary to study the topic of power system network reconfiguration, which is of great theoretical and practical significance for reducing power system failure and outage losses and ensuring safe, rapid and intelligent restoration of power supply after accidents.

In this paper, based on the basic principle and model of particle swarm optimization, mutation operation and tent mapping are introduced, and an improved particle swarm optimization algorithm for power system network reconfiguration is proposed. The algorithm combines the definition of population diversity, discrete particle swarm optimization, inertia weight, and convergence factor selection and dynamic update to improve and optimize the classical particle swarm optimization algorithm. The improved particle swarm optimization (PSO) algorithm is used to simulate the power network system reconfiguration, and the experimental results are compared with the other two algorithms.

The experimental results show that the improved particle swarm optimization algorithm has faster calculation speed, better convergence speed, and convergence accuracy in power system network reconfiguration; the calculation results have small randomness and good stability and have excellent performance compared with the other two algorithms. The improved particle swarm optimization algorithm can provide solutions for power system network reconfiguration and better solve the problems in power system network reconfiguration, which provides ideas and evidence for the next research in this field.
Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Y. Tan and K. Ding, “A survey on GPU-based implementation of swarm intelligence algorithms,” Annals of Thoracic and Cardiovascular Surgery: Official Journal of the Association of Thoracic and Cardiovascular Surgeons of Asia, vol. 11, no. 9, pp. 391–396, 2015.

[2] S. Cheng, Q. Zhang, and Q. Qin, “Big data analytics with swarm intelligence,” Industrial Management & Data Systems, vol. 116, no. 4, pp. 646–666, 2016.

[3] M. Mahi, Ö. K. Baykan, and H. Kodaz, “A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem,” Applied Soft Computing, vol. 30, pp. 484–490, 2015.

[4] Y. J. Gong, J. J. Li, Y. Zhou et al., “Genetic learning particle swarm optimization,” IEEE Transactions on Cybernetics, vol. 46, no. 10, pp. 2277–2290, 2017.

[5] N. K. Jain, U. Nangia, and J. Jain, “A review of particle swarm optimization,” Journal of the Institution of Engineers, vol. 99, no. 4, pp. 1–5, 2018.

[6] M. Schmitt and R. Wanka, “Particle swarm optimization almost surely finds local optima,” Theoretical Computer Science, vol. 561, pp. 57–72, 2015.

[7] R. Y. Harold and M. Rajaram, “Energy-aware multipath routing scheme based on particle swarm optimization in mobile ad hoc networks,” The Scientific World Journal, vol. 2015, Article ID 284276, 9 pages, 2015.

[8] B. Yao, B. Yu, P. Hu, J. Gao, and M. Zhang, “An improved particle swarm optimization for carton heterogeneous vehicle routing problem with a collection depot,” Annals of Operations Research, vol. 242, no. 2, pp. 303–320, 2016.

[9] X. Liang, W. Li, Y. Zhang, and M. Zhou, “An adaptive particle swarm optimization method based on clustering,” Soft Computing, vol. 19, no. 1, pp. 431–448, 2015.

[10] F. Marini and B. Walczak, “Particle swarm optimization (PSO) a tutorial,” Chemometrics and Intelligent Laboratory Systems, vol. 149, pp. 153–165, 2015.

[11] M. Chihi, “Self-adaptive check and repair operator-based particle swarm optimization for the multidimensional knapsack problem,” Applied Soft Computing, vol. 26, pp. 378–389, 2015.

[12] M. R. Bonyadi and Z. Michalewicz, “Particle swarm optimization for single objective continuous space problems: a review,” Evolutionary Computation, vol. 25, no. 1, pp. 1–54, 2017.

[13] Y. Li, X. Bai, L. Jiao, and Y. Xue, “Partitioned-cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation,” Applied Soft Computing, vol. 56, pp. 345–356, 2017.

[14] L. Wang, H. Geng, P. Liu et al., “Particle swarm optimization based dictionary learning for remote sensing big data,” Knowledge-Based Systems, vol. 79, pp. 43–50, 2015.

[15] S. Chatterjee, S. Sarkar, S. Hore, N. Dey, A. S. Ashour, and V. E. Balas, “Particle swarm optimization trained neural network for structural failure prediction of multistoried RC buildings,” Neural Computing and Applications, vol. 28, no. 8, pp. 2005–2016, 2017.

[16] F. Shabbir and P. Omenzetter, “Particle swarm optimization with sequential niche technique for dynamic finite element model updating,” Computer-Aided Civil and Infrastructure Engineering, vol. 30, no. 5, pp. 359–375, 2015.

[17] Z. Beheshti and S. M. Shamsuddin, “Non-parametric particle swarm optimization for global optimization,” Applied Soft Computing, vol. 28, pp. 345–359, 2015.

[18] Z. Beheshti, S. M. Shamsuddin, and S. Hasan, “Memetic binary particle swarm optimization for discrete optimization problems,” Information Sciences, vol. 299, pp. 58–84, 2015.

[19] R. P. Singh, V. Mukherjee, and S. P. Ghoshal, “Particle swarm optimization with an aging leader and challengers algorithm for the solution of optimal power flow problem,” Applied Soft Computing, vol. 40, pp. 161–177, 2016.

[20] Z. Chen, R. Xiong, and J. Cao, “Particle swarm optimization-based optimal power management of plug-in hybrid electric vehicles considering uncertain driving conditions,” Energy, vol. 96, pp. 197–208, 2016.

[21] F. Kuang, S. Zhang, Z. Jin, and W. Xu, “A novel SVM by combining kernel principal component analysis and improved chaotic particle swarm optimization for intrusion detection,” Soft Computing, vol. 19, no. 5, pp. 1187–1199, 2015.

[22] J. Rada-Vilela, M. Johnston, and M. Zhang, “Population statistics for particle swarm optimization: single-evaluation methods in noisy optimization problems,” Soft Computing, vol. 19, no. 9, pp. 2691–2716, 2015.

[23] S. Guelc and H. Kodaz, “A novel parallel multi-swarm algorithm based on comprehensive learning particle swarm optimization,” Engineering Applications of Artificial Intelligence, vol. 45, pp. 33–45, 2015.

[24] H. Banka and S. Dara, “A Hamming distance based binary particle swarm optimization (HDBPSO) algorithm for high dimensional feature selection, classification and validation,” Pattern Recognition Letters, vol. 52, pp. 94–100, 2015.

[25] J. J. Kim and J. J. Lee, “Trajectory optimization with particle swarm optimization for manipulator motion planning,” IEEE Transactions on Industrial Informatics, vol. 11, no. 3, pp. 620–631, 2017.