Journal of Network Intelligence ©2022 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **7**, Number **2**, May **2022**

Parallel Seagull Optimization Algorithm for Application in Distribution Network Reactive power optimization

Jun-Jie Shang[1a], Trong-The Nguyen[1,2b], Lyu-Chao Liao[1c], Lingping Kong[3d], Jeng-Shyang Pan[4*]

[1]College of Computer Science and Mathematics,
Fujian University of Technology, Fuzhou 350118, China
[2]Multimedia Communications Lab,University of Information Technology, VNU-HCM, Vietnam
[3] VSB – Technical University of Ostrava, Czech Republic
[4]College of Computer Science and Engineering,
Shandong University of Science and Technology, Qingdao 266590, China
[a]1215443831@qq.com, [b]thent@uit.edu.vn, [c]fjachao@gmail.com, [d]lingpingkong2007@163.com
Corresponding author: *jengshyangpan@gmail.com

Received April 2021; revised July 2021

Abstract. *A newcomer algorithm called the Seagull optimization algorithm (SOA) has got much paid attention; still, it has some disadvantages, e.g., easy falling into the local optimum and slowly converges. This paper proposes a new approach called PSOA based on parallel strategy and adapting spiral shape parameters to increase particles' diversity for improving algorithm performance. The simulation section, the selected benchmark functions, and a reactive power optimization problem are used for evaluating the proposed algorithm performance. The experimental results are compared with the original and other algorithms in the literature, e.g., genetic algorithm (GA), seagull optimization algorithm (SOA), and particle swarm algorithm (PSO). The single-objective and multi-objective simulations of the IEEE30-node system is used as practical application to check the proposed approach. Combining the distribution network's characteristics and a distribution network reactive power is modeled mathematically to reduce its active power loss and increase voltage stability margin under constraining the state variables by using the feasible region principle. The results show that the proposed PSOA has a more vital global search ability and faster convergence speed in solving the distribution network's reactive power optimization problem.*
Keywords: Distribution Network; Reactive power optimization; Multi-objective problem; Feasible region principle; Parallel seagull optimization algorithm

---

1. **Introduction.** With the people's ever-increasing needs for a better life, the electricity demand is constantly increasing [1]. Therefore, the scale of the country's power system is also expanding [2]. However, at present, the country's urban distribution network still has many problems such as lagging construction, poor reliability, and unsatisfactory grid power indicators [3,4]. In response to this situation, using power resources, reducing network losses, improving the quality of power supply, and improving system operation economics is very important [5]. As an essential link in the power system that directly faces users, the distribution network is the key to ensuring the power supply's quality and the system's safe and economical running [3,6]. According to have a given topology,

load, and other distribution network parameters, reactive power optimization means different optimization algorithms [7]. One of the traditional methods' main drawbacks is the computational effort required to globally optimize the large-scale active power problem [8].

To overcome this situation, different alternatives based on meta-heuristic algorithms (MA) have been developed excellently [9,10]. MA are optimization techniques that consider different operators and heuristic rules to explore a bounded search space [11,12]. The regulations and operators of MA are commonly inspired by various natural behavior. There are two major categories of MA algorithms in the related literature, namely Evolutionary and Swarm algorithms. It is possible to find the genetic algorithms (GA) [13], differential evolution (DE) [13] in the evolutionary group. Meanwhile, the swarm-based techniques include particle swarm optimization (PSO) [14], multi-verse optimizer (MVO) [15] and Seagull optimization algorithm (SOA) [16].

The active power problem can be formulated by combining the distribution network's characteristics and a distribution network reactive power. A modeled mathematically is used to reduce its active power loss and increase voltage stability margin under constraining the state variables by using the feasible region principle as optimization problems then, MA can be useful tools. The SOA algorithm [16] is an MA newcomer algorithm that has paid much attention; still, it has some disadvantages, e.g., easy to fall into the local optimum and slowly converging.

This paper proposes a new approach called PSOA based on parallel strategy and adapting spiral shape parameters to increase particles' diversity for improving algorithm performance. The simulation section, the selected different functions, and a reactive power optimization problem are used for evaluating the suggested algorithm performance. The single-objective and multi-objective simulations of the IEEE30-node system [17] is used as practical application to check the proposed approach. The consequences show that PSOA has a more vital global search ability in solving the distribution network's reactive power optimization problem.

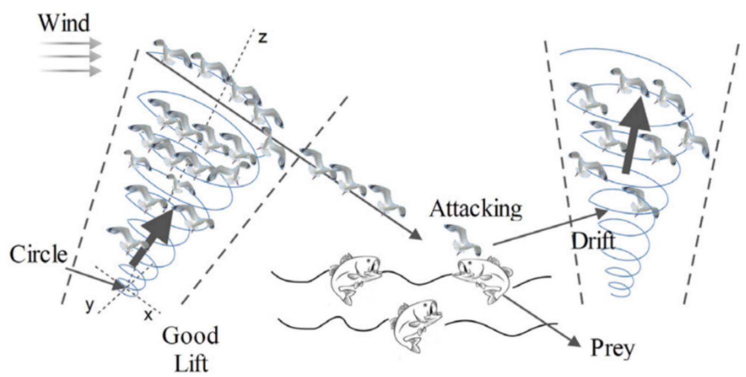

FIGURE 1. Schematic diagram of seagull migration and attack behavior

2. **Reactive power optimization model of distribution network.**

2.1. **Objective function.** The reactive power optimization problem [18] is nonlinear, and the general mathematical model of the nonlinear programming problem is expressed as follows.

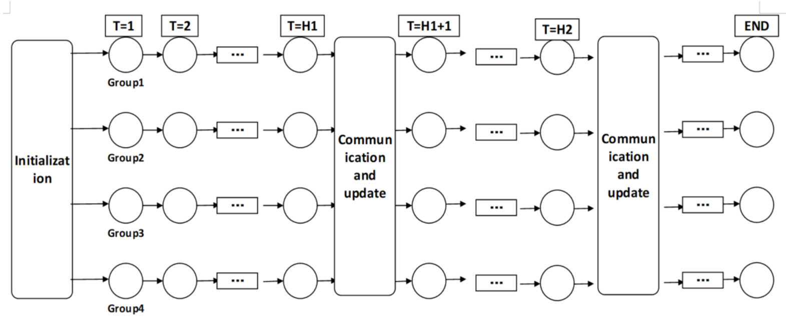

FIGURE 2. An example of parallel communication strategy

$$\begin{cases} minF\left(u,x\right) \\ s.t.g_i\left(u,x\right)=0,\ \ i=1,2,\ldots,m \\ h\left(u,x\right)\le 0, i=1,2,\ldots,m \end{cases} \tag{1}$$

Where u is the control variable. $F(u,x)$ is the objective function of reactive power optimization, which can be considered from various angles such as economy, safety, stability, etc. It can be a target optimization plan or a multi-objective optimization plan. $h(u,x) \le 0$ is an inequality constraint; that is, control variables and state variables must meet the upper and lower limits limits of operation. When the active power in the distribution network is known, and the system and generator operating constraints are met, a multi-objective function is established that minimizes the system power loss and maximizes the static voltage stability margin [2-5].

$$\begin{cases} minF_1 = minP_{loss} = \sum\limits_{K-1}^{N} G_K\left(i,j\right)\left[V_i^2+V_j^2-2V_iV_j\left(\delta_i-\delta_j\right)\right] \\ minF_2 = min\left(\frac{1}{\delta_{min}}\right) \end{cases} \tag{2}$$

Where N is the number of branches; $G_K(i,j)$ is the conductance of line $ij; V_i, V_j$ is the voltage amplitude of nodes i and j; δ_i, δ_j are the voltage phase angles of nodes i, j. δ_{min} is the least singular value of the system convergence power flow Jacobean matrix. In the multi-objective reactive power optimization model, weighting cannot be performed directly because of the different dimensions of the sub-objective functions. To make different sub-objective functions comparable, the objective function can be normalized.

$$\begin{cases} F_1^{'} = F_1/F_0 \\ F_2^{'} = F_2/\left(1/\delta_0\right) \end{cases} \tag{3}$$

In the multi-objective reactive power optimization model, weighting cannot be performed directly because of the different dimensions of the sub-objective functions. To make different sub-objective functions comparable, the objective function can be normalized.

$$\begin{cases} F_1^{'} = F_1/F_0 \\ F_2^{'} = F_2/\left(1/\delta_0\right) \end{cases} \tag{4}$$

Where F_0 is the original active power loss of the system; δ_0 is the least singular value of the original Jacobian matrix of the system. The constraints of the model are divided

Table 1. The pseudo-code of FRP

```
while H_sum(X_i) = 0 do
    if  H_sum(X_j)=0
        f(X_i) < f(X_j) then
            print X_i
        else
            print X_j
        end if
    end while
    while H_sum(X_i) > 0 and H_sum(X_j) > 0 do
        if  H_sum(X_i) < H_sum(X_j)        then
            print X_i
        end if
end while
```

into equality and inequality constraints [6-8]:

$$\begin{cases} P_{Gi} - P_{Di} - V_i \sum\limits_{j=1}^{n} V_j \left(G_{ij} \cos \delta_{ij} + B_{ij} \sin \delta_{ij} \right) = 0 \\ Q_{Gi} - Q_{Di} + V_i \sum\limits_{j=1}^{n} V_j \left(G_{ij} \sin \delta_{ij} - B_{ij} \cos \delta_{ij} \right) = 0 \end{cases} \tag{5}$$

Where V_i, V_j are the voltage amplitudes of the i, j nodes; P_{Gi}, Q_{Gi} are the active output and reactive power output of the generator $node_i$ separately; P_{Di}, Q_{Di} are the active load and reactive load of the nodei separately; G_{ij}, B_{ij} are the real and imaginary parts of the elements in the Number i row and j column of the system admittance matrix separately; δ_{ij} is the voltage phase angle between $node_i and node_j$; n is the quantity of nodes directly connected to $node_i$.

$$\begin{cases} T_t^{min} \le T_t \le T_t^{max} \\ V_g^{min} \le V_g \le V_g^{max} \\ Q_c^{min} \le Q_c \le Q_c^{max} \end{cases} \tag{6}$$

$$\begin{cases} V_i^{min} \le V_i \le V_i^{max} \\ Q_g^{min} \le Q_g \le Q_g^{max} \end{cases} \tag{7}$$

Where T_t is the tap position of the on-load tapping transformer; T_t^{max}, T_t^{min} are the maximum and minimum tap positions of the on-load tapping transformer; V_g is the voltage amplitude of the generator terminal; V_g^{min}, V_g^{max} are the maximum and minimum values of the generator terminal voltage; Q_c is the reactive power compensation ability; Q_c^{max}, Q_c^{min} are the upper and lower limits of the reactive power compensation ability. V_i is the each load node's voltage amplitude; V_i^{max}, V_i^{min} are the maximum and minimum voltage amplitudes of each load node; Q_g is the reactive power injected by each generator; Q_g^{min}, Q_g^{max} are the maximum and minimum values of reactive power injected by each generator [9,10]. In this article, a weighted method is used to transform the multi-objective solution of reactive power optimization into a single-objective problem.

$$minf = \sum_{i=1}^{n} \omega_i f_i \tag{8}$$

Where ω_i is a weighting factor, and its value is determined according to the relationship of the multi-objective problem to be sought, reflecting the trade-off relationship between economy and voltage stability in the reactive power optimization problem; n is the number of objective functions, where, $n = 2, \omega_1 + \omega_2 = 1$. The larger the value of ω_1, the optimization plan focuses on the economy; the larger the value of ω_2, the more focus on the voltage stability margin of the system.

2.2. **Constraint handling.** Since the control variable itself is the search space of the algorithm, no additional constraints are required. Load node voltage and generator reactive power injection power are state variables. Usually, the constraint of state variables is generally adopts the penalty function method. However, selecting the penalty factor in the penalty function is a highly complicated and challenging process. Therefore, this position proposes a feasible region principle (Feasible region principle-FRP). This method judges whether the current position is out of range by comparing the relationship between the current state variable value and the constraint condition. The expression is expressed as follows.

$$H_{sum}\left(X_i\right) = \sum_{i=1}^{N_1} M\left(V_i\right) + \sum_{i=1}^{N_2} M\left(Q_g\right) \tag{9}$$

$$minf = \sum_{i=1}^{n} \omega_i f_i \tag{10}$$

Where $H_{sum}(X_i)$ is the evaluation function of the state variable, and X_i out of bounds can be obtained, N_1, N_2 are the number of state variables; $M(V_i), M(Q_g)$ indicate the limit value of the constraint of the present position; X_i is presented as follows.

$$M\left(C\right) = \begin{cases} C_{min} - C, \ C < C_{min} \\ 0, C_{min} \leq C \leq C_{max} \\ C - C_{max}, C > C_{max} \end{cases} \tag{11}$$

When the value of $H_{sum}(X_i)$ is 0, it means that X_i is within the constraint range. During the operation of the algorithm, the appropriate solution is comprehensively selected by comparing the $H_{sum}(X_i)$ value and the fitness value of the objective function. The specific pseudo code as shown in Table 1.

The above is the case where the feasible region principle X_i is better than X_j, that is, the value of the objective function is compared when the two are not out of bounds, and individuals with excellent fitness values are retained; When one party crosses the boundary, weed out the individual who crossed the boundary; In the case that both are out of bounds, the value of the state variable evaluation function is compared, and the smaller individual is retained. This method does not require an additional selection of empirical coefficients, which can reduce the algorithm's solution time to a certain extent.

3. **Seagull Optimization Algorithm.**

3.1. **Theory of seagull optimization algorithm.** The most important characteristics of seagulls are migration and attacking behavior [16]. Migration is due to the change of seasons. Seagulls move from one place to another on a large scale to find a better living environment. In the process of migration, to avoid collisions between seagulls, there will be certain differences in their flying positions. Seagulls can constantly change their flying position to move toward the best position in the entire seagull population [16]. Attack refers to the process in which seagulls attack fish and shrimps through a spiral-shaped flight while searching for food [16]. Figure 1 shows a schematic diagram of seagull migration and attack behavior. The SOA's basic idea is to perform a global search through migration behavior, perform a local search through attack behavior, and iterate continuously to find the optimal solution.

A. Migration (Global Search)

The algorithm realizes a global search by imitating seagull groups' migration process from one place to another. At this stage, seagulls should meet 3 conditions:

(1) Avoid collision

To prevent collisions between seagulls, the algorithm calculates its updated new position by adding variable A. The expression is presented as follows.

$$C_s(x) = A \times P_s(x) \tag{12}$$

Where $C_s(x)$ represents the new position of the seagull after migration, which is not inconsistent with other seagull positions; $P_s(x)$ represents the current position of the seagull, x represents the current iteration number, and A represents the motion behavior of the seagull in a given search space. It is expressed as follows.

$$A = f_c - (x \times (f_c/Max_{iteration})) \tag{13}$$

Where $x = 0, 1, 2, \ldots,\ Max_{iteration}$, f_c can control the frequency of variable A, reducing the value of A linearly from 2 to 0; $Max_{iteration}$ represents the maximum number of iterations of the algorithm.

(2)The direction of the optimum position

To avoid conflicts with other seagulls' positions during the movement, the seagulls will move in the direction of the best position. The expression is given as follows.

$$M_S(x) = B \times (P_{bs}(x) - P_s(x)) \tag{14}$$

Where $M_S(x)$ represents the direction of the best position; $P_{bs}(x)$ represents the best position of the seagull; B is a random number, which mainly plays a role in balancing global and local search. $B = 2 \times A^2 \times random$, and $random$ is a random number in the range of [0,1].

(3) Close to the optimum position

After the seagull moves to a position where it will not collide with other seagulls, it moves in the direction of the best position to reach a new position. The expression is presented as follows.

$$D_S(x) = |C_S + M_S| \tag{15}$$

where $D_S(x)$ is the new position of the seagull.

B. Attack (Local Search)

Seagulls in the migration process maintain the best relationship between the height of the attack by the movement of the wings and body weight. When prey is found, the seagull will attack the prey in a spiral motion by constantly changing the angle and speed of its attack. The motion behavior in the x, y, and z planes is described as follows:

$$x' = r \times \cos(k) \tag{16}$$

Table 2. Selected benchmark functions in experiments

No.	Functions	Dim	Space solution
F1	$\sum_{i=1}^{n} x_i^2$	30	[-100,100]
F2	$\sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[-10,10]
F3	$\sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j \right)^2$	30	[-100,100]
F4	$max_i\{\lvert x_i \rvert, 1 \leq i \leq n\}$	30	[-100,100]
F5	$\sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 - (x_i - 1)^2]$	30	[-30,30]
F6	$\sum_{i=1}^{n} (x_i + 0.5)^2$	30	[-100,100]
F7	$\sum_{i=1}^{n} ix_i^4 + random[0,1)$	30	[-100,100]
F8	$\sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	30	[-500,500]
F9	$\sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[-5.12,5.12]
F10	$-20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)} + 20 + e$	30	[-32,32]
F11	$\frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x[i]}{\sqrt{i}}\right) + 1$	30	[-600,600]
F12	$4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	30	[-50,50]
F13	$\left(\frac{1}{500} + \sum_{j=1}^{25} \left(j + \sum_{i=1}^{2} (x_i - a_{ij})^6\right)^{-1}\right)^{-1}$	30	[-50,50]

$$y' = r \times \sin(k) \tag{17}$$

$$z' = r \times k \tag{18}$$

$$r = u \times e^{kv} \tag{19}$$

where r is the radius of spiral, k is a random angle value in the range of $[0, 2\pi]$, $0 \leq k \leq 2\pi$. u and v are related constants of the spiral shape, and the value is generally 1. e is the base of the natural logarithm. The attack position of the seagull is expressed as follows:

$$P_s(x) = (D_s \times x' \times y' \times z') + P_{bs}(x) \tag{20}$$

where $P_s(x)$ is the attack position of the seagull.

Parallel strategy

To effectively make up for the shortcomings of the SOA, we introduce the concept of multiple groups that can maintain particles' diversity to ensure that as many optimal solutions as optimization process [3,19,20]. The probability of jumping out of the local optimal solution be improved and large-scale parallel computing can be realized [21,22]. The specific operation is as follows: First, the entire population is grouped to construct a parallel processing structure, and several sub-populations are obtained. Then each sub-population evolves independently according to the iteration rules. After reaching the

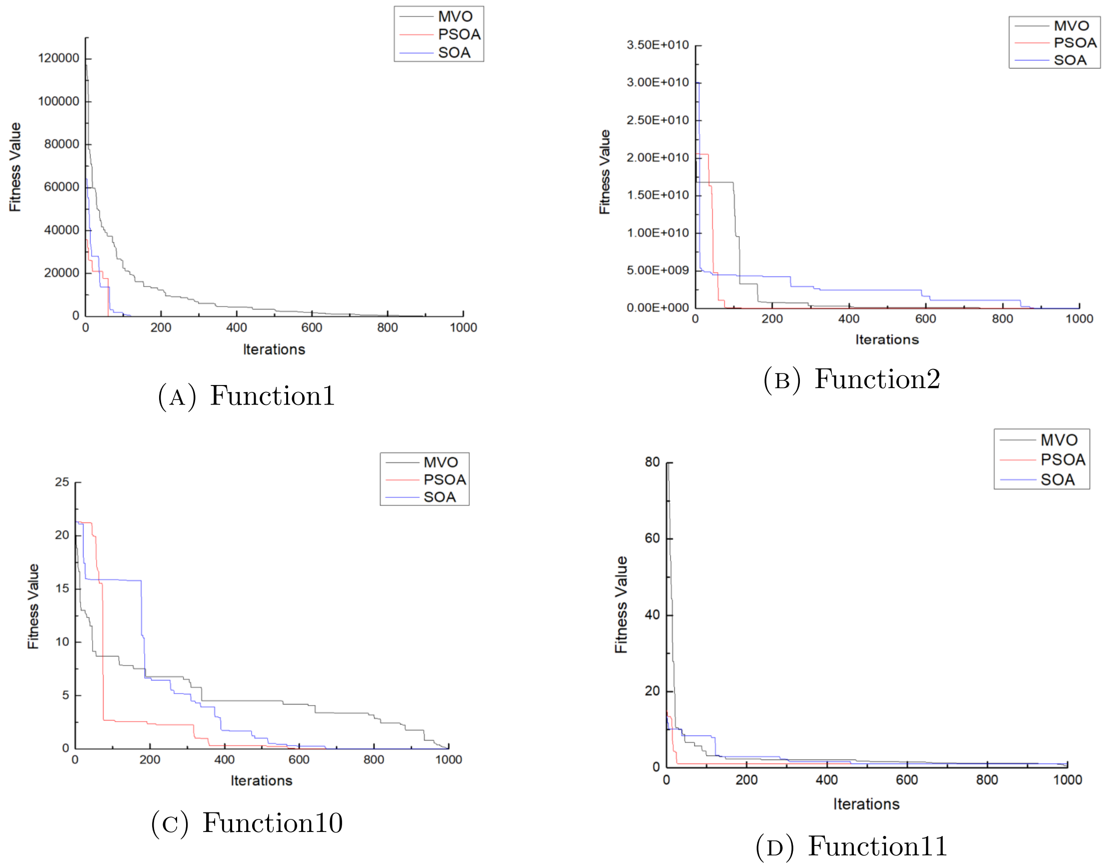

FIGURE 3. Comparison of the optimization obtained values for functions of F1,F2,F10, and F11

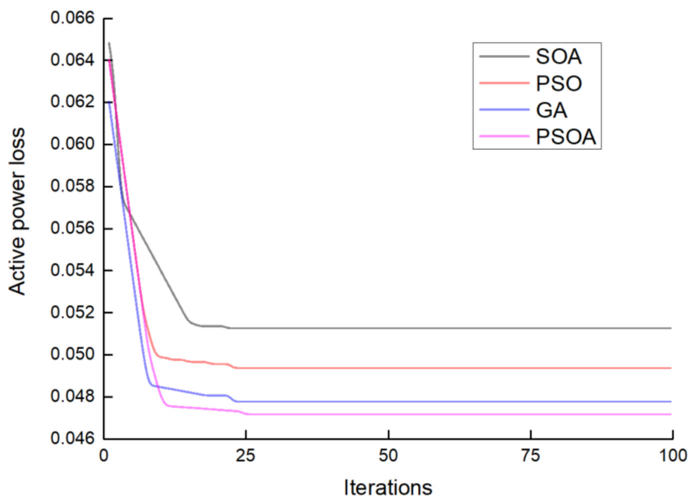

FIGURE 4. Convergence comparison diagram of IEEE-30 node system algorithm

predetermined number of iteration, the inter-group communication strategy is triggered to exchange information between different groups, accelerate the flow of information between groups, and increase the diversity of the particles.

Table 3. Performance for MVO, SOA and PSOA under benchmark functions

Funs.	MVO			SOA			PSOA		
	Best	Worst	Mean	Best	Worst	Mean	Best	Worst	Mean
F1	6.25968	14.6782	9.94788	1.183353	2.633811	1.782539	0.80384	2.450165	1.41395
F2	0.471061	129.0723	12.41392	6.356714	9.871342	7.836592	0.162052	2.648086	1.00422
F3	34.89919	266.7548	98.29809	1.056968	3.027458	2.013695	0.099983	1.736716	1.38486
F4	7.343796	25.55794	16.25627	0.486397	0.739518	0.613259	0.028982	0.632065	0.23605
F5	203.9623	3937.377	870.7867	193.6741	417.3596	289.5497	48.63713	96.52189	56.1559
F6	1.534871	3.587659	2.205965	0	3.4158	1.05963	0	1.460632	0.77385
F7	0.070347	0.214607	0.135718	1.875485	13.14572	6.158572	4.23E-05	0.01193	0.00258
F8	11.34357	29.11692	17.43693	1.035914	2.827901	1.687224	0.385079	0.81924	0.53216
F9	168.1375	423.5731	255.3015	13.981	97.29306	44.36149	34.72547	67.74048	46.8780
F10	2.1785	4.5365	2.8756	1.381629	2.396161	1.935747	0.431522	1.379993	0.71586
F11	1.015532	3.059278	2.035383	0.021307	0.075328	0.046472	0.010791	0.020815	0.01358
F12	2.447	13.524	6.38799	0.143193	0.396382	0.237621	0.160846	0.286414	0.19064
F13	0.745501	16.70711	11.35456	2.495443	4.879254	3.670828	0.853762	3.586647	1.76049

Table 4. IEEE30 node control variables and active power loss optimization results

Variable	*SOA*	*PSO*	*GA*	*PSOA*
UG2	0.9980	1.0954	1.0953	1.0940
UG5	1.0403	1.0739	1.0769	1.0747
UG8	1.0160	1.0730	1.0793	1.0771
UG11	0.9585	1.0603	1.0922	1.0906
UG13	1.0090	1.0910	1.0901	1.0912
T1	0.9750	1.0500	1.0750	1.0000
T2	1.0750	1.1000	1.0250	1.0250
T3	0.9500	1.0750	0.9750	1.0750
T4	1.1000	1.0500	0.9500	1.0250
QC1	0.4000	0.4000	0.4000	0.4000
QC2	0.0600	0.0800	0.0800	0.0800
Ploss	0.0512	0.0493	0.0477	0.0471
Loss reduction rate	***6.3%***	***9.63%***	***12.68%***	***13.8%***

In PSOA, the initialized seagull population is divided into 4 groups, and each group of seagulls evolves with the increase of the number of iterations in the original SOA algorithm. After reaching the predetermined number of iteration H_1, H_2, the optimal global solution P^t is used to replace the individual with the worst fitness in each group. Figure 2 shows this communication strategy in the form of a flowchart.

Adaptive spiral shape parameter

In the original seagull optimization algorithm, seagulls' attack behaviors represent the local exploration of the algorithm [23,24]. The spiral radius r is an important variable for coordinating the algorithm's exploration and development capabilities. A larger r in the early stage is beneficial to increase the diversity of particles, and a smaller r in the later stage is taken value is conducive to the local development of the algorithm; however, the parameter u that determines the value of r in the original algorithm is a fixed value, which leads to insufficient diversity of seagulls in the early stage and reduces the exploration ability of the algorithm [25,26]. Simultaneously, the convergence of the algorithm is inadequate when an accurate local search is required in the later stage [27].

In order to ensure the better early global exploration and later local development capabilities of PSOA, u should maintain a relatively large value in the early iteration of the iteration and continue to decrease with the update of the algorithm. According to the convergence principle, the following three u functions are defined [23,28].

$$u = u_{max} - l \times \left(\frac{u_{max} - u_{min}}{L}\right) \tag{21}$$

$$u = (u_{max} - u_{min}) \times \left(1 - \left(\frac{l}{L}\right)^{\delta}\right)^{1/\delta} + u_{min} \tag{22}$$

$$u = u_{max} - \log\left(1 + q \times e^{\alpha - \frac{\beta}{l/L}}\right) \tag{23}$$

where u_{max} and u_{min} are the maximum and minimum values of u. Formula 21 is a linearly decreasing function; u decreases linearly with iteration. Formula 22 is a parabolic decreasing type, and u decreases convexly with iteration (that is, first decreases slowly and then accelerates to decrease); The coefficient $\delta \geq 1$, when the coefficient $\delta = 1$, formula 22 can be derived as formula 21. Formula 23 is an inverted S-curve model; after logarithmic and transformation, the inverse S-shaped decreasing type is obtained, u is convex first and then concave decreasing with iteration, and the constants q and β are both numbers greater than $0, \alpha \in R$. In the SOA algorithm, r controls the range of seagull attacks. Therefore, in order to enhance the position information interaction and algorithm iteration optimization performance of the PSOA algorithm, u is updated using the inverse S-shaped decreasing function of formula 23, and the range of u value is $(0, 1)$,and $q = 65, \beta = 0.7894, \alpha = -2, u_{max} = 1$.

Therefore, r in this article is expressed explicitly as follows:

$$r = \left[u_{max} - \log\left(1 + q \times e^{\alpha - \frac{\beta}{l/L}}\right)\right] \times e^{kv} \tag{24}$$

Where: l represents the current number of iterations; L represents the maximum number of iterations of the algorithm.

3.2. **Algorithm performance test.** In this section, 13 test functions are used to evaluate the performance of the PSOA algorithm [29]. Among the 13 test functions, 7 are unimodal functions and 6 are multimodal functions. The unimodal function has only one global optimal solution and no local trap (optimal local solution), so it can be used to test the convergence speed of the algorithm. The multimodal function has a globally optimal solution and has one or more local optimal solutions. This feature can be used to test the ability of the algorithm to avoid falling into the optimal local solution. The 13 test functions are shown in the following Table 2:

Some algorithms, e.g., MVO [15], SOA [16] are used to compare with the proposed PSOA. For the experimental results to be convincing, we test each algorithm 20 times.

Table 5. Reactive power optimization results of IEEE-30 system

Numbering	*Active power losses*	*Voltage stability margin*
1	0.0483	0.1387
2	0.0484	0.1395
3	0.0485	0.1404
4	0.0486	0.1405
5	0.0489	0.1409
6	0.0494	0.1414
7	0.0497	0.1417
8	0.0501	0.1421
9	0.0512	0.1422

Table 6. Solutions for different optimization goals

Types of system	*Active power loss*	*Voltage stability margin*
Minimum active power loss	0.0483	0.1387
Maximum voltage stability margin	0.0512	0.1422

The number of iterations of all algorithms is set to 1000, the number of dimensions is 30, and the initial population is 80. Take the best solution, the worst solution, and the average value for comparison. Simultaneously, select $F1, F2, F10, F11$ to draw a convergence curve and compare the convergence trend and stability of the three algorithms.

It can be seen from Table 3 that the PSOA algorithm is better than the other two algorithms in terms of the optimization accuracy of these 13 benchmark functions. Compared with the SOA algorithm, the PSOA algorithm has won 9 times in the performance test of these 13 functions, which is 2 times worse and 2 times similar performance. From the "optimal" point of view, the PSOA algorithm has achieved the better performance 10 times; From the "worst" point of view, the PSOA algorithm has achieved the better performance 9 times and 3 times similar performance; From the "mean" point of view, the PSOA algorithm achieves the better performance 10 times and similar performance 2 times.

Figure 1 shows the convergence curve of the optimal values of $F1, F2, F10$, and $F11$; from the results, the PSOA algorithm has good convergence speed and convergence accuracy and has apparent advantages in optimization performance.

4. **Application For Distribution Network Reactive power optimization.** This article is aimed at a simple IEEE-30 node system [17], in which all data are expressed in per unit value, the power reference is 100MW, and the upper and lower limits of the bus voltage are 1.1 and 0.9. The algorithm is verified by MATLAB2018b simulation. There are a total of six generator nodes in the IEEE30 node system. Among them, node 1 is a balance node, 2, 5, 8, 11, 13 are PV nodes, and the remaining nodes are all PQ nodes. The upper and lower limits of the generator terminal voltage are 1.1 and 0.9; four transformer branches 6-9, 6-10, 4-12, and 28-27; the upper and lower limits of the transformation ratio of the two parallel capacitor compensation nodes 10 and 24 are 1.1 and 0.9, and the

TABLE 7. Multi-objective reactive power optimization results of IEEE-30 node test system

Objective function	*Number of particles*	*Initial value*	*Maximum*	*Minimum*
Active power loss	***80***	***0.546***	***0.0512***	***0.0483***
Voltage stability margin	***80***	***0.1213***	***0.1422***	***0.1387***

upper and lower limits of the parallel capacitor adjustment are 0.5 and 0.1. The IEEE-30 node system has 11 control variables: adjustable transformers T_1, T_2, T_3, T_4, compensation capacitors Q_{C1}, Q_{C2}, and generator terminal voltages $U_2, U_5, U_8, U_{11}, U_{13}$, Control variables $u = [U_2, U_5, U_8, U_{11}, U_{13}, T_1, T_2, T_3, T_4, Q_{C1}, Q_{C2}]$.

4.1. **Single objective test with active power loss smallest.** Algorithm parameter setting: In this section, set the number of seagull populations $N = 80$, the value of v in the PSOA is 1 the value of v [30] and u in the SOA [16] is 1, and the GA algorithm [13,31] calls MATLAB's GA toolbox. The recombination crossover probability is 0.7, and the PSO algorithm learning factor $c_1 = 1$,$c_2 = 2$,$\omega_{max} = 0.8$,$\omega_{min} = 0.2$, and the maximum number of iterations is 100. To avoid accidental events, the experiment was carried out independently for each algorithm 30 times under the same conditions, and the experimental results were displayed in the standard unit value. The average optimization results of the four algorithms PSO [25], SOA [16], PSOA are carried out, and the convergence comparison chart of the four algorithms is given in Figure 4.

It can be seen from Table 4 that in the IEEE-30 system, with the same initial settings, the system network loss before the optimization is 0.0546. The resulting through the parallel seagull optimization algorithm (PSOA) active power loss is 0.0471, active power loss reduction rate of 13.8%, The SOA reduces the network loss by 6.3%, the PSO reduces the network loss by 9.63%, and the GA reduces the network loss by 12.68%. It shows that the performance of PSOA is significantly improved compared to SOA; Compared with GA [13,30], PSO [25] and SOA [13], PSOA have higher calculation accuracy.

It can be seen from Figure 4 that the PSOA has converged in 26 iterations, converges faster, and can search for the optimal solution with a greater probability. From this analysis, it can be concluded that the PSOA is better than the other three algorithms in solving the single-objective reactive power optimization problem.

4.2. **Multi-objective test with the smallest active power loss and the largest static voltage stability margin.** Algorithm parameter setting: The PSOA in this part is the same as Case 1. Since different objective functions, weighting factors can lead to different operation result, under the premise of ensuring $\omega_1 + \omega_2 = 1$, the larger the value of ω_1 , the more economical the optimization plan is; the larger the value of ω_2 , the more emphasis is on the voltage stability margin of the system. To reflect the different requirements of the optimization scheme on the economy and voltage stability, this experiment selects 9 groups of weight factor combinations for testing: the first group: $\omega_1 = 0.9$ $\omega_2 = 0.1$; the second group: $\omega_1 = 0.8$, $\omega_2 = 0.2$; and the ninth group: $\omega_1 = 0.1$ $\omega_2 = 0.9$.

From the above optimization results, we can get: The relationship between the active power loss of the system and the static voltage stability margin is mutually restrictive and contradictory. It is impossible for different objective function values to reach the optimal value at the same time. The solution obtained in the IEEE30-node system effectively maintains the diversity of understanding, so in actual engineering, it is necessary to choose the optimal solution for its own problem according to different requirements, when tend economy, an optimization solution with less active power loss should be selected; when you tend to be safe, you can choose a solution with a larger static voltage stability margin.

Compared with the initial active power loss of the IEEE-30 system is 0.0546, the initial voltage stability margin of the IEEE-30 system is is 0.1213, after PSOA optimization, the maximum loss reduction rate is 11.5%, the minimum loss reduction rate is 7.8%, and the static voltage stability margin is increased by 17.23%. PSOA has certain advantages in solving the problem of reactive power optimization in the distribution network.

5. **Conclusions.** This paper proposed a new approach called PSOA based on parallel strategy and adapting spiral shape parameters to improve algorithm performance to solve reactive power distribution networks' optimization. The proposed method's evaluation has been implemented by testing the selected benchmark functions and single-objective and multi-objective simulations of the IEEE30-node system. The testing results were compared with the other intelligent algorithms in the literature, e.g., seagull optimization algorithm (SOA), genetic algorithm (GA), and particle swarm algorithm (PSO). The comparative results show that the proposed PSOA provides higher convergence accuracy and a more vital global search ability in solving the distribution network's reactive power optimization problem.

REFERENCES

[1] T.-T. Nguyen, M.-J. Wang, J.-S. Pan, T. Dao, and T.-G. Ngo, A Load Economic Dispatch Based on Ion Motion Optimization Algorithm, *Advances in Intelligent Information Hiding and Multimedia Signal Processing*, vol.157, pp. 115-125, 2020.

[2] R. R. Shoults and D. T. Sun, Optimal power flow based upon PQ decomposition, *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-101, issue 2, pp. 397 - 405, 1982.

[3] C. F. Tsai, T. K. Dao, T. S. Pan, T. T. Nguyen, and J. F. Chang, Parallel bat algorithm applied to the economic load dispatch problem, *Journal of Internet Technology*, vol. 17, no. 4, pp. 761–769, 2016.

[4] S. C. Chu, T. K. Dao, J. S. Pan, and T. T. Nguyen, Identifying correctness data scheme for aggregating data in cluster heads of wireless sensor network based on naive Bayes classification, *EURASIP Journal on Wireless Communications and Networking,* vol. 2020, 52, 2020.

[5] T. K. Dao, T. S. Pan, T. T. Nguyen, and S. C. Chu, Evolved Bat Algorithm for solving the Economic Load Dispatch problem, *Advances in Intelligent Systems and Computing,* vol. 329, pp. 109-119, 2015.

[6] T. T. Nguyen, J. S. Pan, and T. K. Dao, A novel improved bat algorithm based on hybrid parallel and compact for balancing an energy consumption problem, *Information,* vol. 10, no. 6, 194, 2019.

[7] J.-S. Pan, T.-T. Nguyen, T.-K. Dao, T.-S. Pan, and S.-C. Chu, Clustering Formation in Wireless Sensor Networks: A Survey, *Journal of Network Intelligence*, vol. 2, no. 4, pp. 287–309, 2017.

[8] T.-T. Nguyen, J.-S. Pan, T.-Y. Wu, T.-K. Dao, and T.-D. Nguyen, Node Coverage Optimization Strategy Based on Ions Motion Optimization, *Journal of Network Intelligence,* vol. 4, no. 1, pp. 1–9, 2019.

[9] J. S. Pan, P. C. Song, S. C. Chu, and Y. J. Peng, Improved compact cuckoo search algorithm applied to location of drone logistics hub, *Mathematics,* vol. 8, no. 3, 333, 2020.

[10] J.-S. Pan, J. Zhuang, H. Luo, and S.-C. Chu, Multi-group Flower Pollination Algorithm Based on Novel Communication Strategies, *Journal of Internet Technology*, vol. 22, no. 2, pp. 257–269, 2021.

[11] T.-T. Nguyen, H.-J. Wang, T.-K. Dao, J.-S. Pan, T.-G. Ngo, and J. Yu, A Scheme of Color Image Multithreshold Segmentation Based on Improved Moth-Flame Algorithm, *IEEE Access,* vol. 8, pp. 174142–174159, 2020.

[12] T.-T. Nguyen, H.-J. Wang, T.-K. Dao, J.-S. Pan, J.-H. Liu, and S.-W. Weng, An Improved Slime Mold Algorithm and Its Application for Optimal Operation of Cascade Hydropower Stations, *IEEE Access,* vol. 8, pp. 226754 - 226772, 2020.

[13] M. Srinivas and L. M. Patnaik, Genetic Algorithms: A Survey, *Computer (Long. Beach. Calif).*, vol. 27, no. 6, pp. 17–26, 1994.

[14] Y. Shi and R. Eberhart, A modified particle swarm optimizer, *in 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence* (Cat. No.98TH8360), pp. 69–73, 1998.

[15] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, Multi-Verse Optimizer: a nature-inspired algorithm for global optimization, *Neural Computing and Applications,* vol. 27, no. 2, pp. 495–513, Feb. 2016.

[16] G. Dhiman and V. Kumar, Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems, *Knowledge-Based Systems*, vol. 165, pp. 169–196, 2019.

[17] Y. Yang, R. Zhou, and X. Ran, Robust optimization with box set for reactive power optimization in wind power integrated system, *in 2012 IEEE Power and Energy Society General Meeting,* pp. 1–6, 2012.

[18] Y. Zhang, S. Wang, and G. Ji, A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications, *Mathematical Problems in Engineering,* vol. 2015, 931256, 2015.

[19] J.-S. Pan, Z. Meng, S.-C. Chu, and H.-R. Xu, Monkey King Evolution: an enhanced ebb-tide-fish algorithm for global optimization and its application in vehicle navigation under wireless sensor network environment, *Telecommunication Systems: Modelling, Analysis, Design and Management*, vol. 65, no. 3, pp. 351-364, 2017.

[20] J. F. Chang, S. C. Chu, J. F. Roddick, and J. S. Pan, A parallel particle swarm optimization algorithm with communication strategies, *Journal of Information ence and Engineering*, vol. 21, no. 4, pp. 809–818, 2005.

[21] T.-T. Nguyen et al., A hybridized parallel bats algorithm for combinatorial problem of traveling salesman, *Journal of Intelligent and Fuzzy Systems*, vol. 38, no. 5, pp. 5811–5820, Feb. 2020.

[22] L. Kong, J.-S. Pan, V. Snášel, P.-W. Tsai, and T.-W. Sung, An energy-aware routing protocol for wireless sensor network based on genetic algorithm, *Telecommunication Systems volume*, vol. 67, no. 3, pp. 451–463, 2018.

[23] A. N. K. Nasir, M. O. Tokhi, O. Sayidmarie, and R. M. T. R. Ismail, A novel adaptive spiral dynamic algorithm for global optimization, *in 2013 13th UK Workshop on Computational Intelligence (UKCI),* pp. 334–341, 2013.

[24] J.-S. Pan, T.-L.-P. Nguyen, T.-G. Ngo, T.-K. Dao, T.-T.-T. Nguyen, and T.-T. Nguyen, An Optimizing Cross-Entropy Thresholding for Image Segmentation based on Improved Cockroach Colony Optimization, *Journal of Information Hiding and Multimedia Signal Processing,* vol. 11, no. 4, pp. 162–171, 2020.

[25] Z. Meng and J. Pan, QUasi-affine TRansformation Evolutionary (QUATRE) algorithm: The framework analysis for global optimization and application in hand gesture segmentation, *in 2016 IEEE 13th International Conference on Signal Processing (ICSP),* pp. 1832–1837, 2016.

[26] S. Zhang, X. Su, X. Jiang, M. Chen, and T.-Y. Wu, A Traffic Prediction Method of Bicycle-sharing based on Long and Short term Memory Network., *Journal of Network Intelligence,* vol. 4, no. 2, pp. 17–29, 2019.

[27] X. Xue, J. Chen, and D. Chen, Matching Biomedical Ontologies Through Compact Hybrid Evolutionary Algorithm, *Journal of Information Hiding and Multimedia Signal Processing,* vol. 10, no. 1, pp. 110–117, 2019.

[28] T. T. Nguyen, J. S. Pan, and T. K. Dao, An Improved Flower Pollination Algorithm for Optimizing Layouts of Nodes in Wireless Sensor Network, *IEEE Access*, vol. 7, pp. 75985–75998, 2019.

[29] J. J. Liang, B. Y. Qu, P. N. Suganthan, and A. G. Hernández-Díaz, Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization, URL:https://al-roomi.org/multimedia/CEC_Database/CEC2013/RealParameterOptimization/ CEC2013_RealParameterOptimization_TechnicalReport.pdf

[30] L. Kang, R. Chen, N. Xiong, Y. Chen, Y. Hu, and C. Chen, Selecting Hyper-Parameters of Gaussian Process Regression Based on Non-Inertial Particle Swarm Optimization in Internet of Things, *IEEE Access*, vol. 7, pp. 59504–59513, 2019.

[31] T. Shukla, S. Singh, and K. Naik, Allocation of optimal distributed generation using GA for minimum system losses in radial distribution networks, *International Journal of Engineering Science and Technology,* vol. 2, no. 3, pp. 94–106, 2010.