Research on BDS/GPS Integrated Navigation Satellite Selection Algorithm Based on Particle Swarm Optimization

Ershen Wang, Chaoying Jia, Tao Pang, Pingping Qu
and Zhixian Zhang

Abstract In the multi-constellation satellite navigation system, all the visible satellites are used for positioning, which will increase the computation amount of the receiver and affect the real-time positioning. How to quickly and effectively select visible satellites for positioning is a research topic. For this problem, a satellite selection algorithm based on Particle Swarm Optimization (PSO) is proposed. In this method, the visible satellite is numbered, random grouping, and each group as a particle; the velocity-displacement model in the PSO keeps the particles gradually close to the minimum value of the GDOP. Under a series of simulation experiments, the key parameters such as inertia weight factor, acceleration coefficient and maximum velocity of PSO are determined. Besides, local search based on chaos mechanism are introduced, which can avoid the results of PSO algorithm into local optimum. Finally, the performance of satellite selection with PSO is confirmed to be remarkable by the simulation experiments under different numbers of selected satellites. The results show that this method can quickly select satellites under BDS and GPS system, and the result meets receiver positioning accuracy.

Keywords BeiDou navigation satellite system (BDS) · Global positioning system (GPS) · Satellite selection · Particle swarm optimization (PSO)
Geometric dilution of precision

E. Wang (✉) · C. Jia · T. Pang · P. Qu · Z. Zhang
School of Electronic and Information Engineering, Shenyang Aerospace University,
Shenyang, China
e-mail: wanges_2016@126.com

E. Wang
Liaoning General Aviation Key Laboratory, Shenyang Aerospace University,
Shenyang, China

J. Sun et al. (eds.), *China Satellite Navigation Conference (CSNC) 2018 Proceedings*, Lecture Notes in Electrical Engineering 497,
https://doi.org/10.1007/978-981-13-0005-9_59

1 Introduction

With the development of the Global Positioning System (GPS) of the United States, GLONASS of Russia, Galileo of the EU and BeiDou of China, the number of visible satellites has increased significantly, and the combination of constellations will improve the user's satellite geometry. Abundant navigation signal source provides the conditions that can improve the performance of the navigation and positioning, but it also brings signal processing burden of the receiver. If all visible satellites are used for positioning, it will be difficult to guarantee the real-time positioning. Therefore, selecting the best set of visible satellites is necessary from the multiple visible satellites, which can improve the accuracy and real-time performance of navigation and positioning.

The measure standard of satellite selection algorithm is the spatial geometry distribution between users and visible satellites. According to the characterization of the spatial distribution between user and visible satellites, dilution of precision is the most widely used in multiple parameters [1]. Among them, the Geometric Dilution of Precision (GDOP) is related to geometry between user and satellites. The smaller its value, the better the spatial geometric distribution and the higher the positioning accuracy. At present, researchers mainly focus on three aspects: the calculation of GDOP [2, 3], the fast satellites selection algorithm based on geometric distribution [4] and the satellites selection algorithm based on genetic algorithm [5, 6]. All of algorithms above mentioned can reduce the calculation of satellites selection to some extent, and get an effective result of satellites selection in shorter time. Similar to genetic algorithm, particle swarm optimization (PSO) is a bionic algorithm. The system forms the initial population by randomly generating a set of solutions. And then, according to the adaptation information of individuals, the algorithm starts to search among the population. Finally, PSO system will find the optimal solution in the search space. The PSO algorithm is well applied in many fields. This paper attempts to apply the PSO algorithm to satellites selection, and the PSO satellites selection model is established. By introducing a local search based on chaos mechanism, the global search ability of the algorithm is improved. The performance of the PSO was also verified by comparing with the results of satellites selection of traversal method. In the case of satellites selection of BDS/GPS, the number of satellites selection is more than 5, so the PSO satellites selection takes less time and has higher efficiency.

2 Calculation Method on DOP

The position of the navigation receiver position consists of the satellite position $(\mathrm{x}_u, \mathrm{y}_u, \mathrm{z}_u)$ and the pseudo-range ρ_i (j = 1, 2, … n, n is the total of visible satellites received by the receiver at the current time), the pseudo-range observation equation is as follows:

$$\rho_j = \sqrt{(\mathrm{x}_j - \mathrm{x}_u)^2 + (y_j - y_u)^2 + (z_j - z_u)^2} + c(t_u - t_j) + \varepsilon_\rho \tag{1}$$

ρ_j is the value of the pseudo-range observations, c is the speed of the light, t_u is the prior time of the receiver relative to the GPS time, t_j is the prior time that the jth satellite's clock relative to the GPS system, ε_ρ is the observed noise of the pseudo-range (including the ionospheric, tropospheric delay and delay caused by multipath effects).

In the process of positioning, the approximate position of the receiver is assumed to be $\hat{\rho}_j = (\widehat{\mathrm{x}}_u, \widehat{\mathrm{y}}_u, \widehat{\mathrm{z}}_u)$, the estimated clock bias is $\hat{t}_u$. The first-order Taylor series expansion of (1) at the receiver approximate position and the clock bias yields the linearized equation:

$$\Delta\rho_j = -\frac{x_j - \widehat{x}_u}{\rho_0}\Delta x_u - \frac{y_j - \hat{y}_u}{\rho_0}\Delta y_u - \frac{z_j - \widehat{z}_u}{\rho_0}\Delta z_u - c\Delta t_u + \varepsilon_\rho \tag{2}$$

where, $\rho_0 = \sqrt{(\mathrm{x}_j - \widehat{\mathrm{x}}_u)^2 + (y_j - \hat{y}_u)^2 + (z_j - \widehat{z}_u)^2}$, $\Delta\rho_j = \rho_j - \hat{\rho}_j$, $\Delta x_u = x_u - \hat{x}_u$, $\Delta y_u = y_u - \hat{y}_u$, $\Delta z_u = \mathrm{z}_u - \hat{\mathrm{z}}_u$, $\Delta t_u = t_u - \hat{t}_u$.

$l_i = -\frac{x_j - \hat{x}_u}{\rho_0}$, $m_i = -\frac{y_j - \hat{y}_u}{\rho_0}$, $n_i = -\frac{z_j - \widehat{z_u}}{\rho_0}$, respectively represent the direction cosine of the jth satellite unit vector from the approximate position. Equation (2) can be written in matrix form:

$$\Delta\rho = H\Delta X \tag{3}$$

where,

$$H = \begin{bmatrix} l_1 & m_1 & n_1 & 1 \\ l_2 & m_2 & n_2 & 1 \\ \cdots & \cdots & \cdots & \cdots \\ l_n & m_n & n_n & 1 \end{bmatrix}, \Delta X = \begin{bmatrix} \Delta x_u \\ \Delta y_u \\ \Delta z_u \\ -c\Delta t_u \end{bmatrix}, \Delta\rho = \rho_j - \hat{\rho}_j \tag{4}$$

The typical implementation of the solution algorithm is an iterative linear least-squares method, assuming that the pseudo-range error is $d\rho$ [1], and the components of the same distribution and independent of each other, the mean squared error equal to the satellite user range error, that is:

$$\mathrm{cov}(d\rho) = I_{n\times n}\sigma^2_{UERE} \tag{5}$$

So,

$$\mathrm{cov}(dX) = (H^T H)^{-1} H^T \mathrm{cov}(d\rho) H (H^T H)^{-1} = (H^T H)^{-1} \sigma_{UERE}^2 \tag{6}$$

The dilution of the precision can be defined by the symmetric matrix $(H^T H)^{-1}$, the geometric dilution of the precision GDOP can be defined as the square root of the covariance matrix, and the other commonly used performance indicators are HDOP (Horizontal Dilution of Precision), PDOP (Position Dilution of Precision), TDOP (Time Dilution of Precision).

$$GDOP = \sqrt{\mathrm{trace}\{(H^T H)^{-1}\}} \tag{7}$$

For BDS/GPS dual constellation, due to the different range error performance, it is usually done by appending extra columns in Eq. (4).

$$H = \begin{bmatrix} l_{G,1} & m_{G,1} & n_{G,1} & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{G,m} & m_{G,m} & n_{G,m} & 1 & 0 \\ l_{B,1} & m_{B,1} & n_{B,1} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{B,n} & m_{B,n} & n_{B,n} & 0 & 1 \end{bmatrix} \tag{8}$$

In Eq. (8), the subscript G represents the GPS constellation, the number of visible satellites is $1\ldots m$, subscript B represents the BDS constellation, the number of visible satellites is $1\ldots n$. GDOP is still calculated using Eq. (7), which is a function only related to the satellite and user geometry layout. The value of GDOP changes with the change of the number of satellites involved in navigation, and decreased monotonously with the increase of satellites. The details are given in [7].

DOP is an important basis for satellites selection algorithm. The smaller the DOP is, the higher the accuracy of positioning resolution is. The satellites selection according to DOP can ensure the positioning accuracy [1].

3 Particle Swarm Optimization Algorithm

Particle Swarm Optimization (PSO), as a hot research topic in many fields, has been successfully applied in multiple fields (including neural network training, power system, communication, image processing, etc.) [8]. The PSO algorithm is a stochastic optimization algorithm based on swarm intelligence. it mimics the foraging behavior of birds. The problem search space is analogized to the flight space of birds, and each bird is abstracted into a particle to represent a candidate solution of the problem, the optimization needs to find the optimal solution is equivalent to looking for food. As a group optimization algorithm proposed by the simulation of biological predation in nature, the PSO algorithm can solve the optimization problem through the cooperation and competition among individuals in the population.

3.1 Basic Principle of PSO Algorithm

PSO is initialized to a random set of random particles (random solutions). In each iteration, the particle updates itself by tracking two "extrema" [9–11]. The first one is the optimal solution found by the particle itself, and called the individual extrema (pbest). The other is the optimal solution found by the entire population, and called the global extrema (gbest). After a limited number of iterations, each particle gradually approaches the optimal solution, and eventually finds the optimal solution. Taking the *i*th particle in d-dimensional search space as an example, its position and velocity are expressed as $X_i = [x_{i1}, x_{i2}, \ldots, x_{id}]$, $V_i = [v_{i1}, v_{i2}, \ldots, v_{id}]$.

The fitness value of each particle is evaluated by the fitness function, and the best position (pbest) $P_i = [p_{i1}, p_{i2}, \ldots, p_{id}]$ after t iterations, and the best position (gbest) found in the population. Then, PSO algorithm updates velocity and position of each particle by follow equations:

$$v_{i,j}(t+1) = \omega v_{i,j}(t) + c_1 r_1 [p_{i,j} - x_{i,j}(t)] + c_2 r_2 [p_{g,j} - x_{i,j}(t)] \tag{9}$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1), \quad j = 1, \ldots, d \tag{10}$$

where, ω is the inertial weight factor, c_1 and c_2 is acceleration coefficient, r_1 and r_2 is a random number that is evenly distributed between 0 and 1. In addition, by setting the velocity limitations $[v_{\min}, v_{\max}]$ and the position range $[x_{\min}, x_{\max}]$ of the particles, the movement of the particles can be appropriately restricted.

Different from genetic algorithm, PSO algorithm uses real number coding, standard PSO algorithm does not have more parameters to be adjusted. The search performance depends on the balance between global search and local improvement, which depends largely on the control parameters of the algorithm, including the size of population, maximum velocity, maximum iteration, inertia weight factor and acceleration coefficient. Standard PSO does not have many parameters to adjust, usually select according to experience [12].

The velocity limitations of particle $V_{\max}$ determine the particle's maximum move distance in an iteration cycle. Shi gives the simulation results of the PSO system optimization experiment, when $V_{\max}$ is the alone value of $[2, V_{\max}]$ and the velocity inertia within [0.1, 1.05].

The acceleration factor c_1 and c_2 is usually chosen as 2, Clerc recommended the value of $(c_1 + c_1)/2$ is 1.494.

Inertia factor ω, the value is generally between (0, 1), Clerc suggested that its value is 0.729, and Riget adopted strategy which decreasing ω with the different number of iteration.

Termination conditions generally choose to reach the maximum cycle setting, or to meet the predetermined error requirements.

3.2 *Chaotic Particle Swarm Optimization Algorithm*

In order to make the particle swarm optimization algorithm faster at the optimal search time and avoid falling into the local optimum, the chaos concept, the time-varying inertia weight and the time-varying acceleration coefficient are introduced in PSO [13]. For a given optimization function, CPSO correspond to the search process as chaotic orbit traversal process, the search process has the ability to avoid local extremum, and finally obtain the global optimal solution or satisfactory solution. Optimization based on chaotic sequence includes two key steps:

(1) Mapping chaos space to the solution space of optimization problem;
(2) The use of chaotic dynamic characteristics to achieve the solution space search.

Logistic maps are one-dimensional discrete-time nonlinear systems and exhibit quadratic nonlinearity [12]. Logistic mapping is expressed as

$$x_{i+1} = f(x_i) = \mu x_i(1 - x_i) \quad i = 0, 1, 2, \ldots \tag{11}$$

where, $x_i \in (0, 1)$, when $x_0 \notin \{0, 0.25, 0.5, 0.75, 1\}$ and $\mu = 4$, the sequence generated by Logistic mapping presents the chaotic dynamic characteristics. Small changes of the initial variables will lead to huge differences in the subsequent tracks.

4 The Steps of PSO Satellites Selection

Suppose at a certain moment, the number of visible satellites is n, and select a certain number of m from them. The GDOP value as small as possible. In the PSO satellites selection, each particle represents a combination of visible satellites, the location of the particle is determined by m elements, each element represents a visible satellite.

(1) Coding

Randomly array all the visible satellites observed by the receiver at the current time, and then sequentially code the visible satellites from $1, \ldots, n$ to the next, and the serial numbers correspond to the visible satellites one by one and remain unchanged in subsequent operations.

(2) Generated initial population randomly

Set the population with M, the initial population of $G_0 = \{g_{0i}\}(i = 1, \ldots M)$, each particle $g_{0i} = [g_{0,1}, g_{0,2}, \ldots, g_{0,\mathrm{m}}]$ is taken from all visible satellites in initial population. In order to make the number of selected satellites is the same, we must ensure that each element is different in g_{0i}. Initialization velocity $v_0 = [v_{0,1}, v_{0,2}, \ldots, v_{0,m}]$, the subscript "0" means that the particles go through 0th iterations, that means the initial position and velocity. After repeated experiments, it is concluded that, in order to avoid the search results of PSO falling into local optimum, a chaotic sequence is introduced into the initial population. First, a random number z_i of (0, 1) is generated, and z_i is updated according to formula (11). the chaotic space is mapped to the solution space to be optimized according to the formula $x_i = x_{i\min} + z_i(x_{i\max} - x_{i\min})$.

(3) Fitness calculation

The fitness used in this paper is the GDOP value which is the combination of visible satellites with encoding, and denoted as the target value $f_{t,i}$ of the particle, and the subscript "t" means that the particle go through tth iterations. The particles in the initial population are substituted into the fitness function in turn, and the fitness value of each particle is obtained. choose the smallest fitness value in population as the initial global optimal position *gbest* and each particle itself position as the initial local optimal position *pbest*.

(4) Updated

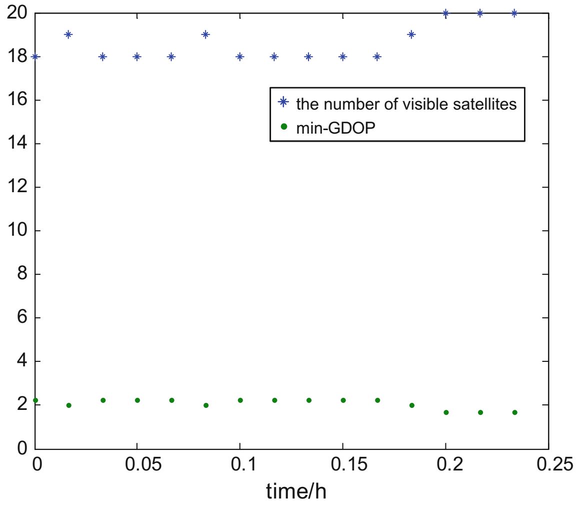
For each particle, according to the position and velocity update formulas of PSO, the position $g_{t,i}$ and the velocity $v_{t,i}$ of the particle in the population are continuously corrected, and the target value corresponding to $f_{t,i}$ is calculated respectively.

5 Experiment and Simulation Analysis

PSO satellites selection is to solve the GNSS constellations, and the number of satellites selected more than 5. This paper select BDS/GPS dual constellations, the receiver coordinates are [−2279827.3156, 5004704.3094, 3219776.2093], the number of satellites selected is 6. The satellite position is calculated by ephemeris, and the cut-off height angle of the satellite is set as 5°. The simulation time start at 2016.7.31.00:00:00, the simulation duration is 900 s, and the simulation step is 60 s.

In this paper, we use the traversal method to get the GDOP value, and choose its value as a reference. Assuming that the number of all visible satellites is n, then select m visible satellites from them. Thus, the number of combinations are C_n^m. The so-called traversal method is to calculate the GDOP values of combinations one by one, and obtains the minimum value of GDOP. The number of BDS/GPS visible satellites and their corresponding minimum GDOP values are shown in Fig. 1.

Fig. 1 The number of BDS/GPS visible satellite and their min-GDOP



As can be seen in Fig. 1, at the same time, the receiver receives about 18 BDS/GPS visible satellites, take selecting 6 visible satellites from 18 visible satellites as an example, it needs to calculate GDOP values C_{18}^{6} = 18,564 times. Each time-consuming about 2.6 s or so.

Set the algorithm parameters according to the PSO satellites selection procedure in Sect. 4: Number of iterations $MaxIt = 50$, Inertia coefficient $\omega = 0.7298$, Acceleration coefficient $c_1 = c_2 = 1.4962$, $V_{\max} = 2$. GDOP value is calculated at a certain moment. As the number of iterations increases, the GDOP value changes as shown in Fig. 2.

By traversing method, the minimum value of GDOP at current moment is 2.25. From the results in the figure, the convergence velocity of PSO algorithm is fast,

Fig. 2 GDOP values change with different

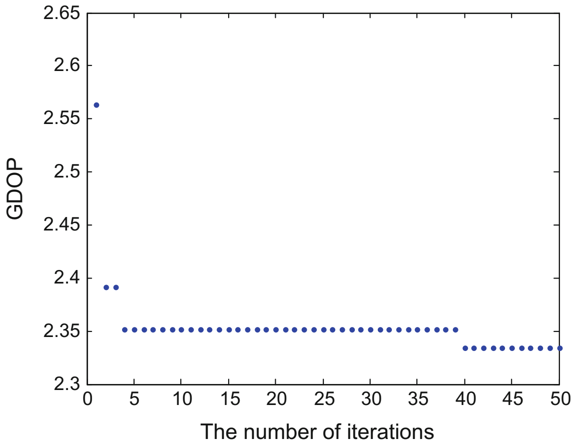
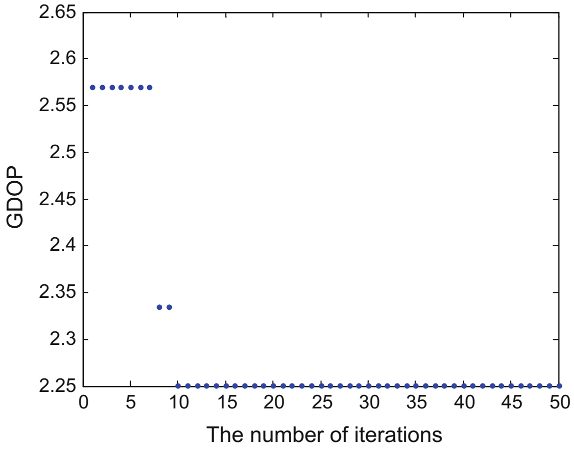


Fig. 3 GDOP values change with different number of PSO iterations number of CPSO iterations



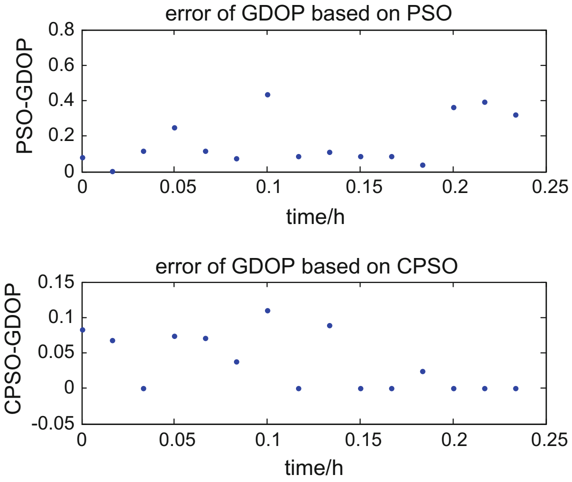
and the number of iterations is stable around 2.34 when it is less than 10th iterations. Obviously, PSO algorithm emerges "premature" phenomenon (that is trapped in the local optimum). The following uses CPSO algorithm, and the introduction of time-varying inertia weight and time-varying acceleration coefficient. Time-varying inertia weight ω usually decrease linearly from 0.9 to 0.4, and are expressed as $\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \times \frac{it}{\mathrm{MaxIt}}$ in the iterative process. $\omega_{\max}, \omega_{\min}$ is the maximum and minimum inertia coefficients, respectively; it, MaxIt is the current iteration number and the maximum iteration number. The maximum number of iterations *MaxIt* is 50. Simulation test at the same moment, the results is obtained in Fig. 3.

It can be seen from this figure that the CPSO algorithm also converges when the number of iterations is less than 10, and effectively improves the disadvantage that PSO satellites selection algorithm fall into the local optimum. When the number of iterations is 50, the time-consuming is 1.512039 s. If the number of iterations is reduced to 15 times, the satellites selection time will be shorter.

The PSO and CPSO is applied to select the satellite, and each result is compared with the results of the traversal method. GDOP calculation error of the new algorithm is shown in Fig. 4.

It can be seen from the figure that the CPSO algorithm has a more stable GDOP error than the standard PSO algorithm. The calculation error of PSO-GDOP is between 0 and 0.5, while the calculation error of CPSO-GDOP is basically stable at 0–0.15.

Fig. 4 The results error of PSO and CPSO select satellites



6 Conclusions

In this paper, we present a method of selecting satellites by the Particle Swarm Optimization (PSO) algorithm. A satellites selection model of PSO is constructed and introducing the concept of chaos, effectively avoids the limitation of the PSO algorithm which is easy to get into the local optimum and improves the system stability. The simulation results show that under the BDS/GPS dual constellations, and the number of satellites selection is more than 5, the satellites selection time of proposed algorithm is shortened to 42.3% than traverse method, and the minimum GDOP calculation error is stable within a certain range, which can quickly select the satellites.

Acknowledgements This study is funded by the National Natural Science Foundation of China (61571309 & 61101161), the Liaoning BaiQianWan Talents Program, Scientific Study Project for Liaoning Province Ministry of Education (L201716), Liaoning Excellent Talents in University (LR2016069) and the Aerospace Science Foundation of Avic (2015ZC54010).

References

1. Zhang J (2004) Air-ground collaborative airspace surveillance. Aviation Industry Press, Beijing
2. Phatak MS (2001) Recursive method for optimum GPS satellite selection. IEEE Trans Aerosp Electron Syst 37(2):751–754
3. Meng F, Wang S, Zhu B (2015) GNSS reliability and positioning accuracy enhancement based on fast satellite selection algorithm and RAIM in multi-constellation. Aerosp Electr Syst Magazine IEEE 30(10):14–27

4. Liu S, Zhao GR, Gao C, Zeng B (2017) A fast satellite selection algorithm for GPS/BD integrated navigation system. Electr Opt Control 24(3):32–35
5. Song D, Xu CD, Hu CS, Zhang PF (2015) Satellite selection with genetic algorithm under multi-constellation. J Astronaut 36(3):300–308
6. Huo HY, Zhang XL (2015) Fast satellite selection method for integrated navigation systems. J Beijing Univ Aeronaut Astronaut 41(2):273–282
7. Han TX (2014) Research on GNSS multi-system satellite selection strategy. Shanghai Jiaotong Univ 03:28–45
8. Wang L, Liu B (2008) Particle swarm optimization and scheduling algorithm. Tsinghua University Press, Beijing
9. Poli R, Kennedy J, Blackwell T (2007) Particle swarm optimization: an overview. Swarm Intell 1(1):33–57
10. Eberhart RC, Shi Y (2002) Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 congress on evolutionary computation, 2001, vol 1. IEEE, pp 81–86
11. Jiang Y, Hu T, Huang CC et al (2017) An improved particle swarm optimization algorithm. Appl Mech Mater 193(1):231–239
12. Pan F, Li WX, Gao Q (2013) Particle swarm optimizer and multi-object optimization. Beijing Institute of Press, Beijing
13. Kuru L, Ozturk A, Kuru E et al (2015) Determination of voltage stability boundary values in electrical power systems by using the chaotic particle swarm optimization algorithm. Int J Electr Power Energy Syst 64(15):873–879