A satellite selection algorithm based on adaptive simulated annealing particle swarm optimization for the BeiDou Navigation Satellite System/Global Positioning System receiver

Ershen Wang¹,2, Caimiao Sun¹, Chuanyun Wang¹, Pingping Qu¹, Yufeng Huang¹ and Tao Pang¹

Abstract
In this article, we propose a new particle swarm optimization–based satellite selection algorithm for BeiDou Navigation Satellite System/Global Positioning System receiver, which aims to reduce the computational complexity of receivers under the multi-constellation Global Navigation Satellite System. The influences of the key parameters of the algorithm—such as the inertia weighting factor, acceleration coefficient, and population size—on the performance of the particle swarm optimization satellite selection algorithm are discussed herein. In addition, the algorithm is improved using the adaptive simulated annealing particle swarm optimization (ASAPSO) approach to prevent converging to a local minimum. The new approach takes advantage of the adaptive adjustment of the evolutionary parameters and particle velocity; thus, it improves the ability of the approach to escape local extrema. The theoretical derivations are discussed. The experiments are validated using 3-h real Global Navigation Satellite System observation data. The results show that in terms of the accuracy of the geometric dilution of precision error of the algorithm, the ASAPSO satellite selection algorithm is about 86% smaller than the greedy satellite selection algorithm, and about 80% is less than the geometric dilution of precision error of the particle swarm optimization satellite selection algorithm. In addition, the speed of selecting the minimum geometric dilution of precision value of satellites based on the ASAPSO algorithm is better than that of the traditional traversal algorithm and particle swarm optimization algorithm. Therefore, the proposed ASAPSO algorithm reduces the satellite selection time and improves the geometric dilution of precision using the selected satellite algorithm.

Keywords
BeiDou Navigation Satellite System, Global Positioning System, satellite selection, particle swarm optimization, adaptive simulated annealing

Introduction
Multi-constellation Global Navigation Satellite System (GNSS) increases the number of visible satellites, which improves the positioning performance. Better satellite
geometry and more navigation-redundant information can be obtained by the multi-constellation GNSS than by a single satellite navigation system; however, the signal-processing burden of a GNSS receiver is also higher. In addition, the reception of satellite signals is affected by factors such as the atmosphere of the earth, building obscuration, and electromagnetic wave interference, which adversely affect the receiver positioning performance. However, for multi-constellation GNSS, if all-in-view satellites are used for positioning solution, the data processing process of the receiver will be more complicated and take more time. In addition, in dual-constellation navigation, only five satellites at least are needed to solve the positioning calculation. Therefore, it is necessary to study a fast and effective satellite selection algorithm, select a better subset of satellites from all-in-view satellites for receiver positioning, and improve the real-time performance of positioning while ensuring the accuracy of navigation and positioning. Generally, the geometric dilution of precision (GDOP) is used to evaluate positioning accuracy. The smaller the GDOP, the higher the positioning accuracy corresponding to the satellite combination. Therefore, the common satellite selection algorithm is based on the minimum GDOP. However, the GDOP value does not always decrease linearly with the increasing of the number of visible satellites. After the number of visible satellites reaches a certain threshold, the GDOP value will not be significantly improved. Therefore, the satellite selection algorithm has become an important research area for GNSS receiver based on multi-constellation integrated navigation. The calculation method of GDOP is as follows. It can be seen from the formula that the calculation of GDOP requires the matrix’s multiplication and inversion

\[
\text{GDOP} = \sqrt{\text{trace}(H^T H)^{-1}}
\]  

(1)

where \( H \) represents the observation matrix in the geodetic coordinate system.

Previous research found the existing satellite selection algorithm can be applied when more than four satellites are available and focuses on two aspects of research. First, the area is divided according to the satellite elevation angle and azimuth; then, the visible satellite subset is selected. Second, based on the minimum GDOP traversal satellite selection algorithm, all satellites are grouped according to the chosen number of satellites, the GDOP for each set of satellites is calculated, and the minimum GDOP and its corresponding satellite combination are searched, the large number of GDOP calculations make the satellite selection process too time-consuming. In response to this problem, in recent years, the calculation process of GDOP is simplified and an optimization algorithm is adopted to reduce the number of calculations of the GDOP. In 2010, a satellite selection method based on the genetic algorithm (GA) was proposed; GA reduces the computation time of GDOP using its own rapid optimization capabilities. However, GA satellite selection falls easily into local optimum. This algorithm has been greatly improved; furthermore, the accuracy of the satellite selection results has been improved. However, the GA algorithm has too many parameters that require adjustment during the calculation process. Therefore, even though GA can guarantee the accuracy of the satellite selection, as the number of visible satellites increases, the satellite selection speed of GA decreases. In 2018, a satellite selection method using the signal levels of multi-constellation GNSS was proposed. The aforementioned algorithms can reduce the complexity of the satellite selection calculation to a certain extent and improve the effectiveness of the satellite selection results.

To fully utilize the advantages offered by multi-constellation GNSS, it is necessary to study a fast and effective satellite selection algorithm. In this article, in order to reduce the computational burden of the GNSS receiver and the time-consuming of satellite selection, the particle swarm optimization (PSO) algorithm is combined with the satellite selection; second, the influence of PSO algorithm parameters on the performance of satellite selection is analyzed to select reasonable parameter values. In addition, in order to ensure the accuracy of the satellite selection results, the adaptive simulated annealing (ASA) is introduced to adapt the global optimal satellite selection of the BeiDou Navigation Satellite System (BDS)/Global Positioning System (GPS) dual-constellation.

This article is organized as follows. The analysis of related work is specifically described in section “Related work,” and some common satellite selection algorithms are instructed to prove the validity of the research content in this article. In section “The basic PSO algorithm,” the PSO algorithm is introduced, and the influence of its parameters on the PSO satellite selection algorithm is analyzed. In section “ASAPSO satellite selection algorithm,” the satellite selection algorithm based on ASA is mainly described. Section “Simulation experiment and result analysis” describes the relationship between GDOP and the number of satellites selected, and through a large number of simulation experiments, it focuses on analyzing the results of the PSO satellite selection algorithm and the improved PSO satellite selection algorithm. The conclusion and future research of this article are described in section “Conclusion.”
Related work

Comprehensive analysis of the above research results, GDOP is an important indicator of satellite selection. The traditional traversal satellite selection algorithm completes the satellite selection by selecting a subset of satellites with the smallest GDOP value. With the increasing of the number of visible satellites in the space domain, the calculation of the traversal satellite selection algorithm becomes more complicated. In addition, calculating the GDOP of a large number of satellite subsets involves matrix inversion and multiplication. Based on the above description, it can be seen that the traversal satellite selection algorithm is no longer applicable to the situation where the number of satellites selected under multiple constellations exceeds five.

Dan and El-Sherief.18 discussed the application of neural network algorithms in satellite selection and analyzed the advantages of neural network-based satellite selection; Phatak19 proposed a satellite selection algorithm based on the revolving door method and the matrix inversion lemma, which completed the satellite selection through recursive calculation; Meng et al. proposed a fast satellite selection algorithm for multi-constellation. This method is based on both Newton’s identities for GDOP fast computation and optimal satellite geometric distribution for less cycle participation, and the GDOP of proposed method is very close to optimal result.20 These satellite selection algorithms are based on the calculation of the minimum GDOP value, which can meet the rapid satellite selection in a single constellation to a certain extent. However, as the number of satellites increases, the number of satellite combinations also becomes larger and the amount of calculation for selecting satellites increases significantly. These algorithms also have limitations in reducing the amount of calculation, so they can only choose sub-optimal satellite combinations.

In recent years, some new fast and high-precision satellite selection algorithms based on the minimum GDOP value are proposed. L. Jin et al.21 analyzed the relationship between the GDOP of the multi-satellite integrated system and the elevation angle, azimuth angle of the visible satellite, and proposed a negative exponential decay model in which the relative ratio of GDOP before and after the selection increases with the chosen number of satellites, based on the elevation angle of the visible satellite classify all visible satellites to achieve indirect satellite selection. Li et al. proposed a fast-rotating divisional satellite selection algorithm based on the uniform distribution of the sky. The algorithm partitions visible satellites according to the relationship between the number of satellites and GDOP. Simulation experiments show that the rotating partitioned satellite selection algorithm improves the utilization of navigation data and improves the accuracy of satellite selection.22 HU et al. proposed an improved fast satellites selection algorithm. Use the elevation and azimuth information of satellites to divide the area, and finally determine the six satellites selected for positioning by Newton’s iterative GDOP calculation method. Theoretical analysis and experimental results show that, this algorithm greatly reduces the amount of calculation and has higher positioning accuracy at the slight sacrifice of positioning accuracy. It can prove the effectiveness of the algorithm better than the quasi-optimal algorithm.23 In addition, in 2019, Gerbeth et al.24 proposed a Ground-Based Augmentation System (GBAS)-based GDOP satellite selection algorithm to reduce the amount of computation; Xu et al.25 proposed a GDOP satellite selection method based on nondominated sorted genetic algorithm II (NSGA-II) algorithm to improve the real-time performance and reliability of satellite selection. In 2019, a satellite selection strategy based on ambiguity dilution of precision (ADOP) was proposed; this method can greatly reduce the time consumption of the fixed low cutoff elevation angle multi-GNSS.26

These algorithms have greatly improved the real-time performance and accuracy of satellite selection, but there are also problems such as poor stability and many adjustment parameters. Therefore, in this article, an improved PSO satellite selection algorithm is proposed based on the BDS/GPS dual-constellation.

In general, the PSO algorithm is used to solve optimization problems in nonlinear complex systems and has been successfully applied in many fields, such as production scheduling, system control, and image processing, such as: Guo et al.27 proposed a distributed PSO approach, which can optimize the hyperparameters to find high-performing Convolution neural network (CNN); M Wang et al.28 proposed to apply the PSO algorithm to the application of power cables; ES Wang et al.29 proposed to apply the PSO algorithm to the autonomous integrity monitoring algorithm of navigation receivers. By analyzing the existing research, it can be known that the PSO algorithm can greatly improve the real-time performance, but it is difficult to meet the accuracy requirements in places with higher accuracy requirements. Therefore, based on the PSO algorithm, an adaptive simulated annealing particle swarm optimization (ASAPSO) satellite selection algorithm is proposed to improve the speed and real-time performance of satellite selection.

The basic PSO algorithm

PSO algorithm proposed by Kennedy and Eberhart30 to mimic the foraging behavior of birds. In the search
space, individual particles compare the optimal positions they pass with the optimal positions of all the particles and adjust their speed continuously, so that they can move closer to the global optimal position. Individuals are called “particles,” and each particle is a point in the $d$-dimensional space. The position of the $i$th particle can be expressed as $x_i = [x_{i1}, x_{i2}, \ldots, x_{id}]$, where $x_{i1}, x_{i2}, \ldots, x_{id}$ represents the component of the position of the $i$th particle in $d$-dimensional space, respectively. In the process of particle motion, an individual optimal value $pbest_i$ is generated according to the experience of each particle. In addition, a global optimal value $gbest$ is generated for the entire population. Particles constantly adjust their position based on these two “best values,” so that they are constantly approaching the global optimal value. The updated position of the particles is as shown in equations (2) and (3), and the location change velocity of the $i$th particle is expressed as $v_i = [v_{i1}, v_{i2}, \ldots, v_{id}]$

$$v_i(t + 1) = \omega v_i(t) + c_1 r_1 [pbest_i - x_i(t)] + c_2 r_2 [gbest - x_i(t)]$$

$$x_i(t + 1) = x_i(t) + v_i(t + 1), \quad i = 1, \ldots, N$$

where $t$ is the current number of iterations, $N$ is the total number of particles in the population, $\omega$ is the inertia weight, $c_1$ and $c_2$ are the acceleration constants, which are used to adjust the velocity of motion in the $pbest$ and $gbest$ directions, respectively, $r_1$ and $r_2$ are the random numbers uniformly distributed between 0 and 1. The inertia weight plays an important role in the balance between the global and local searches, and its value is usually linearly decremented from 0.9 to 0.4. In the iterative process, $\omega$ is expressed as follows

$$\omega = \omega_{\text{max}} - (\omega_{\text{max}} - \omega_{\text{min}}) \times \frac{t}{\text{Max} t}$$

where $\omega_{\text{max}}$ and $\omega_{\text{min}}$ are the maximum and minimum inertia coefficients, respectively, $t$ and Max $t$ are the current and maximum number of iterations, respectively.

The basic PSO algorithm easily falls into local optimal. In order to overcome the above problem, the concept of simulated annealing was applied in subsequent research, and the effectiveness of the results was improved.

**PSO satellite selection algorithm**

The PSO algorithm is applied to satellite selection. It is necessary to consider the selection of the fitness function, the definition of the particle position, and the method of updating the particle information. The fitness function selected in this article is the GDOP value of each visible satellite combination, which is used to evaluate the spatial geometry of the selected satellite combination. The PSO fast satellite selection algorithm mainly includes the following six steps:

Step 1: based on real data extraction of visible satellites.
Step 2: encoding, the number of each visible satellite is encoded as $1, 2, \ldots, n$.
Step 3: formation of the initial population.
Step 4: selection of the fitness function.
Step 5: update of the particle position and velocity, and fitness function.
Step 6: search for visible satellite subsets with better spatial geometry.

For each particle in the population $x_i$

$$x_i = [x_{i1}, x_{i2}, \ldots, x_{im}]$$

where $x$ is the number of the satellites, $m$ is the number of satellites selected, and $v_i$ is the velocity

$$v_i = [v_{i1}, v_{i2}, \ldots, v_{im}]$$

The initial velocity of the particle is expressed as $[0, 1, 0, 2, \ldots, 0, m]$, examples of initial particles can be expressed as Figure 1.

The particles are iteratively searched according to equations (2) and (3). In the next time step, the velocity of the particle is determined by the following factors: the current time velocity $v_i(t)$, the individual best position $pbest_i$, and the global optimal position $gbest$. Here, $pbest$ is defined as the position corresponding to the minimum fitness function value (GDOP) of the particle during the iterative process and $gbest$ is defined as the particle position corresponding to the minimum fitness function value in the current population. After a finite number of iterations, the particles in the population will converge to a certain value with high probability. Finally, the satellite subset with the smallest GDOP will be obtained. The following Figure 2 shows the flowchart of the PSO star selection algorithm.

**Influence of the parameters on the PSO satellite selection algorithm**

The parameter selection plays a key role in the algorithm performance. The parameters in the PSO fast satellite selection algorithm include the population size, inertia weight, and acceleration coefficients. The PSO algorithm faces a “premature” problem, where in the particle loses diversity, staying at the local optimum, such that the algorithm is unable to find the global optimal solution; consequently, the particle moving speed is too slow to achieve rapid optimization. This problem has been minimized by some researchers to varying
degrees. However, the optimal parameters of the evolutionary algorithm need to be selected based on the specific application and empirical values. Therefore, based on six satellites selected as an example, the effect of parameters on the performance of the algorithm is analyzed based on actual data.

**Influence of inertia weight on the algorithm performance**

The aforementioned conclusion was verified by the experiments of Eberhart and Shi. When $\omega \approx 0.8$, the PSO algorithm has a strong local search ability and can converge to the global optimal solution quickly. When $\omega > 1.2$, the algorithm has a strong global search ability but a slow convergence speed. When $0.8 \leq \omega \leq 1.2$, compared with the two aforementioned cases, the possibility of the algorithm converging to the global optimal solution is greater and the speed of convergence is moderate. Therefore, in this article, the algorithm performance is improved by adaptively changing the inertia weight.

According to the PSO satellite selection procedure, the algorithm simulation parameters were set as follows: the number of iterations was 50; the acceleration coefficients were $c_1 = c_2 = 2$; the population size was $M = 100$; and the inertia weights were 0.4, 0.6, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, and 1.6. The value of each inertia weight was simulated 10 times. The average GDOP value and the average satellite selection time are listed in Table 1.

![Figure 1. A diagram of the initial particle population model.](image1)

![Figure 2. PSO satellite selection algorithm flowchart.](image2)
When the GDOP value is relatively low, the spatial geometry of the satellites is better and the positioning accuracy is higher. Therefore, the fitness function of the algorithm is defined as the GDOP of the computing satellite, and the algorithm quickly searches for the minimum GDOP and its corresponding satellite combination through multiple iterations.

As can be seen from the results in Table 1, the average fitness function obtained by the inertia weight in the range of $0.8 < \omega < 1.2$ is optimal and the average fitness function solution is poor when $\omega < 0.8$ or $\omega > 1.2$. In particular, the global search ability of the particles deteriorates when $\omega$ is large, and the minimum GDOP values are the same in the 10 simulation experiments. This indicates that the algorithm can search a global optimal solution. However, the same epoch containing all-in-view satellites cycles 10 times to find the GDOP minimum value; furthermore, the corresponding satellite selection time is longer. In addition, the average satellite selection time of the PSO algorithm is between 1.5 and 1.7 s, which indicates that the inertia weight has a little effect on the time consumed for satellite selection. Furthermore, the effect of the acceleration coefficient on the algorithm is verified in the following section.

### Effect of the acceleration coefficients on the algorithm performance

The acceleration coefficients $c_1$ and $c_2$ are used to adjust the moving step size of the particles in the optimal directions of the individual and population, respectively. It can be concluded from the particle update equation (1) that when $c_1 > c_2$, the particle update speed depends on a comparison between the particle and the optimal positions; however, when $c_1 < c_2$, the particle update speed tends to compare its position with the optimal position of the population. The number of iterations was set to 50; the inertia weight was $\omega = 0.9$; the population size was $M = 100$; and the acceleration coefficients $c_1/c_2$ were 0.25, 0.5, 1, 2, 3, and 4. The values of each acceleration coefficient were simulated 10 times. The average GDOP value and CPU time results are shown in Table 2.

It can be seen from the results that the preferred combinations of $c_1$ and $c_2$ are (2, 2), (4, 2), and (2, 1). When the combination of $c_1$ and $c_2$ is suitable, the optimization effect of the algorithm is better. In the update iteration process of the algorithm, the choice to use a pair of determined acceleration constant values can
ensure the convergence speed of the algorithm and the search for the global optimal solution.

**Effect of the population size on the algorithm performance**

There are several parameters that need to be adjusted in the basic PSO algorithm. One of these parameters is the size of the population. In general, the population size is selected according to the difficulty of the problem to be solved. In general, a value of 20–50 is common. The number of termination iterations of the first set algorithm was set to 50 generations, the inertia weight was $v = 0.9$, the acceleration coefficients were $c_1 = c_2 = 2$, and the population size $M$ was taken to be 30, 50, 70, 90, 100, 110, 120, 150, 180, and 200. In addition, each of the values of $M$ was simulated 10 times. The average GDOP value and CPU time-consuming results are presented in Table 3.

Table 3 shows that as the population size increases, the average satellite selection time of the algorithm increases. When the population size is $M = 30$, the CPU time occupied by the algorithm is only 0.54 s and the objective function value can be found in 10 iterations. However, the algorithm performance does not increase with increasing population size; first, it decreases and then increases. When the PSO algorithm is used in the fast satellite selection problem, it can be seen that the population size is between 70 and 100, and the average of GDOP is between 2.28 and 2.3. Therefore, the algorithm has its best performance when the population size $M$ is between 70 and 100.

**ASAPSO satellite selection algorithm**

Faced with the application of PSO in the problem of satellite selection, the algorithm parameters were adjusted via simulation experiments. The experimental results show that the parameter selection of the algorithm directly affects the algorithm performance and time spent on satellite selection. Therefore, if the algorithm parameters are fixed, the relationship between the algorithm performance and satellite selection time can only be balanced via a compromise method.

However, the optimization effect of the parameters selected using the compromise method is not ideal and requires that the algorithm parameters be adaptively adjusted with iterative changes. Moreover, when the algorithm enters the iteration, the differences in the particles in the population are very large and the global search ability is strong. However, the positions of the particles in the population gradually approach the global optimal value after multiple search iterations; moreover, the particle search adjustment is slower and global optimal value of the search cannot be guaranteed, resulting in the previously mentioned “premature” problem. In this article, the inertia weight was adaptively adjusted using the target value and PSO algorithm combined with the simulated annealing algorithm, which was used to optimize the satellite selection process to improve the satellite selection accuracy.

**Adaptive inertia weight**

As shown in equation (2), when $\omega \geq c_1$ and $\omega \geq c_2$, the particles are not affected by $p_{best}$ and $g_{best}$, respectively, and they will move according to the original speed; in this case, it is difficult for the particles in the population to converge to a certain value. However, when $\omega \rightarrow 0$, the particle velocity in the next time step is independent of the current velocity and the particles in the population converge rapidly. At this point, the search result obtained by the algorithm is the current population optimal value, rather than the optimal value of the problem solution space, that is, the “premature” problem. Therefore, the algorithm search results are directly affected by the magnitude of the inertia weight $\omega$. In this article, the adaptive inertia weight is changed according to the difference in the particle target value; the update equation is expressed as

| Population size $M$ | Average of GDOP | Maximum value of GDOP | Minimum value of GDOP | Average satellite selection time (s) |
|---------------------|----------------|----------------------|-----------------------|-------------------------------------|
| 30                  | 2.4054         | 2.5099               | 2.2510                | 0.5462                              |
| 50                  | 2.3325         | 2.4196               | 2.2510                | 0.8282                              |
| 70                  | 2.2895         | 2.3645               | 2.2510                | 1.1280                              |
| 90                  | 2.3085         | 2.3585               | 2.2510                | 1.4413                              |
| 100                 | 2.2928         | 2.4196               | 2.2510                | 1.5870                              |
| 110                 | 2.3172         | 2.4196               | 2.2510                | 1.7939                              |
| 120                 | 2.3264         | 2.5402               | 2.2510                | 1.9247                              |
| 150                 | 2.3123         | 2.3707               | 2.2510                | 2.4918                              |
| 180                 | 2.2840         | 2.3346               | 2.2510                | 2.7759                              |
| 200                 | 2.3069         | 2.3707               | 2.2510                | 3.1239                              |

GDOP: geometric dilution of precision.
where $\omega_{\text{max}}$ and $\omega_{\text{min}}$ are the maximum and minimum values of the inertia weight $\omega$, respectively; $f_i$ is the fitness value of the particles; and $f_{\text{avg}}$ and $f_{\text{min}}$ are the average and minimum fitness values of the particles in the population, respectively.40

**PSO algorithm combined with simulated annealing**

Simulated annealing (SA) is an algorithm for solving the combinatorial optimization problem based on the cooling and annealing of metals37,38 and can be used to process cost functions that are nonlinear, discontinuous, and random.39 In the PSO algorithm, the particles will gradually move closer to the best position in the group. When the optimal position of the group is locally optimal and the particle difference in the population is small, all particles will tend to local extreme. Eventually, the output of the algorithm will be inaccurate. SA algorithms are used to provide certain selected probabilities for particles with poor performance. This improves the particle diversity to a certain extent, enhancing the global search ability of the algorithm.40

The algorithm first provides the initial temperature $T_0$. As the temperature decreases, particles with poor performance can be obtained by the algorithm with probability $p$. The relationship between the temperature and probability of obtaining a poor solution can be calculated using the following equation

$$p = \frac{\exp\left(\frac{f_i - f_{\text{opt}}}{T}\right)}{\lambda}$$

where $p$ is the probability of obtaining a poor solution, $f_{\text{opt}}$ is the optimal fitness value of the particles in the population, and $T$ is the temperature. When the temperature is higher, the probability that the algorithm obtains a worse solution will be greater. Therefore, a higher initial temperature is set during the application of the algorithm to improve the global search ability of the particle. However, after a certain ratio of temperature reduction, the search space is gradually reduced until it converges to the global optimal solution.

**ASAPSO satellite selection algorithm**

Assuming that the total number of visible satellites processed by the receiver at the current time is $n$, $m$ satellites with relatively better spatial geometry are selected from the total. The ASAPSO fast satellite selection algorithm mainly includes the following steps (Figure 3).

In Figure 3, $\omega$ represents the adaptive inertia weight; $p$ represents the probability of obtaining a poor solution according to equation (9); and $r$ represents the random number. The flowchart of the ASAPSO satellite selection algorithm can be expressed as Figure 4.

In Figure 4, $f_i$ represents the fitness value of the particles; $f_{\text{avg}}$ represents the average fitness values of the particles in the population; $g_{\text{opt}}$ represents the initial optimal population position; $p_{\text{best}}$ represents the positions of each particle; $\omega$ represents the adaptive inertia weight; $r$ represents the random number; $T$ represents the temperature; and $\lambda$ represents the cooling rate. In the iteration process of the improved algorithm, when the fitness value of the new position of the particle is smaller than the fitness value of its current position, the particle moves to the new position; meanwhile, when the adaptation value of the new position is greater than the current position adaptation value, the particle does not necessarily discard the new position. Instead, it is determined by whether the particle moves to a new location by accepting the difference probability.

**Simulation experiment and result analysis**

**Simulation analysis of the integrated constellation satellite precision factor**

The relationship between the number of satellites combined with BDS, GPS, and BDS/GPS was related to the GDOP factor at the same time using the same observatory. At this time, the receiver coordinates were [-2279827.3156, 5004704.3094, 3219776.2093], and the same clock state is used for GPS and BDS. Meanwhile, the elevation mask angle of satellites was set to 5°, and the simulation duration was 3 h. Figure 5 shows the number of visible satellites under different navigation systems.

Figure 5 shows the number of visible satellites for a single GPS, BeiDou, and BDS/GPS combined constellations. The results show that during the simulation time, the chosen number of satellites in BeiDou and GPS was between 8 and 11 and 8 and 10, respectively. The number of visible satellites in the BDS/GPS
combined constellation twice that of a single constellation; at least 17 satellites can be observed simultaneously.

To determine the relationship between the number of different satellites and their corresponding GDOP, the min-GDOP of the all-in-view satellites and that of different number satellites were solved separately and compared. The results are shown in Figure 6.

Figure 6(a)–(d) shows the min-GDOP value of 6, 8, 10, and 12 satellites selected and the GDOP value of all visible satellites at the same time within 3 h. It can be seen that the more the number of selected satellites, the smaller the value of GDOP and the closer it is to the GDOP value calculated using all the visible satellites. When the number of selected satellites is 10, the curves coincide sometimes. Therefore, it is possible to replace all visible satellites with a certain number of partially visible satellites according to the user’s need for positioning accuracy. When the required positioning accuracy is high, a combination with a larger number of satellites can be selected. And the smaller the GDOP value, the better the positioning accuracy performance. However, as the number of on-demand satellites increases, it takes more time on satellite selection. Therefore, while ensuring the accuracy of satellite selection, it is of great significance to study the fast algorithm for satellite selection.

Simulation analysis of the ASA algorithm

In this article, the computer CPU processor is configured as Intel® Xeon®, 2.20 GHz, and using MATLAB computational language. Real navigation raw data...
were used to verify the performance of the improved algorithm. The navigation message and observation files were obtained from the US IGS website http://www.igs.gnsswhu.cn/index.php/Home/DataProduct/igs.html. The BDS/GPS receiver coordinates were $[2279827.3156, 5004704.3094, 3219776.2093]$, and the chosen number of satellites was 6, 7, 8, and 10. The satellite position was calculated using the navigation ephemeris, the elevation mask angle of the satellites was set to $5^\circ$, and the simulation time was 3 h.

In this article, the parameters of the ASAPSO satellite selection algorithm are given according to multiple experimental tests and related to research. The total number of particles in the initial population $M = 50$; maximum moving speed $v_{\text{max}} = 4$; minimum moving speed $v_{\text{min}} = -4$; acceleration factor $c_1 = c_2 = 1.2$; maximum inertia weight $\omega_{\text{max}} = 0.9$; maximum inertia weight $\omega_{\text{min}} = 0.4$; SA initial temperature $T_0 = 10$. Figure 7 shows the GDOP of the ASAPSO and PSO satellite selection algorithms in the iterative search process.

In the PSO satellite selection algorithm, the particle’s GDOP value remains the same after two iteration times. Obviously, the PSO satellite selection algorithm easily falls into a local optimum. Instead, for the ASAPSO satellite selection algorithm, the particle searched for the global minimum during 50 iterations and the CPU time consumed was 2.736459 s. Compared with the previous experimental data, the traversal algorithm consumes approximately 4 s for one-time satellite selection and the improved algorithm can
achieve fast satellite selection. From the curve in Figure 7, it can be seen that the ASAPSO has the ability to escape local extrema. When the number of iterations is from 15 to 25, the result of the algorithm search is near 2.45; then, in the 26th iteration, the algorithm escapes the local extremum and searches for the global optimal solution.

In the following, the three kinds of algorithm including the greedy, PSO, and ASAPSO satellite selection algorithms are compared with the traversing method. The GDOP calculation error is defined as the difference between the GDOP value obtained by the greedy, PSO and ASAPSO satellite selection algorithms and the GDOP value obtained by the traversing method. The results are represented in Figure 8.

Analyzing Figures 7–11, comparing ASAPSO satellite selection with PSO satellite selection and greedy satellite selection, it can be seen that in the given example of satellite selection, the GDOP error of the ASAPSO satellite selection algorithm is smaller than that of the PSO satellite selection algorithm and the greedy satellite selection algorithm. In addition, with the increasing
of the number of selected satellites, when the number is 10, the GDOP errors corresponding to the three satellite selection algorithms of ASAPSO, PSO and greedy have less close to 0. But the ASAPSO algorithm still has more GDOP errors close to 0.

On the basis of analyzing the accuracy of different algorithms, Table 4 shows the average time required for one-time satellite selection by different satellite selection algorithms.

The data in Table 4 show that ASAPSO takes less time than PSO and traversal, but ASAPSO takes more time than greedy, so it can be seen that greedy and PSO can improve the efficiency of satellite selection, while ASAPSO guarantees the accuracy of satellite selection, it can also improve the speed of satellite selection to a certain extent.

Conclusion
In this work, in order to improve the computational burden of the traditional traversal satellite selection algorithm, this article proposes a new satellite selection model based on the PSO algorithm, which realizes fast satellite selection and also guarantees the GDOP performance. The simulation experiments based on the real BDS/GPS navigation raw data show that: (1) when
the same satellite is selected at the same time, about 86% of the GDOP error of the ASAPSO satellite selection algorithm is less than that of the greedy satellite selection algorithm, and about 80% GDOP error is smaller than that of the basic PSO satellite selection algorithm; (2) ASAPSO algorithm can improve the global search capability of satellite selection; and (3) with the increase in the chosen number of satellites, the time required for each satellite selection algorithm increases, but the time for selecting the minimum GDOP value of satellites based on the ASAPAO algorithm is less than the traversal and PSO algorithms, and higher than the time for greedy algorithm. After a large number of simulation experiments, it is shown that the work proposed in this article can achieve rapid satellite selection for a better satellite combination.

In the future, more computationally efficient algorithms need to be developed for wide usage of our algorithm. In addition, our algorithm can be implemented on the hardware platform for GNSS receiver, which is another interesting topic for future research.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This study was supported by the National Natural Science Foundation of China (71731001 and 61571309), the Talent Project of Revitalization Liaoning (XLYC190702), the Key R & D Projects of Liaoning Province (2020JH2-10100045), the Natural Science Foundation of Liaoning Province (2019-MS-251), the Scientific Research Project of Liaoning Provincial Department of Education (JYT2020142), and the High-Level Innovation Talent Project of Shenyang (RC190030).

ORCID iD
Ershen Wang https://orcid.org/0000-0003-4744-8892

Table 4. Time-consumption comparison of different satellite selection algorithms.

| Number of selected satellites | PSO one-time selection satellite time (s) | ASAPSO one-time selection satellite time (s) | Greedy one-time satellite selection time (s) | Traversal one-time selection satellite time (s) |
|-------------------------------|------------------------------------------|---------------------------------------------|---------------------------------------------|---------------------------------------------|
| 6                             | 1.5495                                   | 1.1299                                      | 0.9587                                      | 4.2853                                      |
| 7                             | 1.8652                                   | 1.2984                                      | 1.0603                                      | 8.6268                                      |
| 8                             | 2.2875                                   | 1.9083                                      | 1.5476                                      | 11.9044                                     |
| 10                            | 2.8185                                   | 2.4728                                      | 1.9877                                      | 19.8327                                     |

PSO: particle swarm optimization; ASAPSO: adaptive simulated annealing particle swarm optimization.

References
1. Ren XD, Chen J, Li XX, et al. Multi-GNSS contributions to differential code biases determination and regional ionospheric modeling in China. *Adv Space Res* 2020; 65(1): 221–234.
2. Li C, Su Z, Li Q, et al. An indoor positioning error correction method of pedestrian multi-motions recognized by hybrid-orders fraction domain transformation. *IEEE Access* 2019; 7: 11360–11377.
3. Bao GZ, et al. Analysis of influence of different satellite combinations on GNSS/INS integrated navigation positioning accuracy. *Sci Surv Mapp* 2019.
4. Xue SQ and Yang Y. Positioning configurations with the lowest GDOP and their classification. *J Geodesy* 2015; 89(1): 49–71.
5. Wang ES, Jia CY, Qu PP, et al. BDS/GPS integrated navigation satellite selection algorithm based on chaos particle swarm optimization. *J Beijing Univ Aeronaut Astronaut* 2019; 45(2): 259–265.
6. Walter T, Blanch J and Kropp V. Satellite selection for multi-constellation SBAS. In: *Proceedings of the 29th international technical meeting of the satellite division of the institute of navigation (ION GNSS + 2016)*, Portland, OR, 12–16 September 2016, pp.1350–1359. Manassas, VA: The Institute of Navigation.
7. Huo HY and Zhang XL. Fast satellite selection method for integrated navigation systems. *J Beijing Univ Aeronaut Astronaut* 2015; 41(2): 273–282.
8. Huang PP, Rizos C and Roberts C. Satellite selection with an end-to-end deep learning network. *GPS Solut* 2018; 22(4): 108.
9. Song D, Xu CD, Hu CS, et al. Satellite selection with genetic algorithm under multi-constellation. *J Astronaut* 2015; 36(3): 300–308.
10. Du H, Hong Y, Xia N, et al. A navigation satellites selection method based on ACO with polarized feedback. *IEEE Access* 2020; 8: 168246–168261.
11. Mosavi MR and Divband M. Calculation of geometric dilution of precision using adaptive filtering technique based on evolutionary algorithms. In: *Proceedings of the IEEE international conference on electrical and control engineering*, Wuhan, China, 25–27 June 2010. New York: IEEE.
12. Xu XF, Ma LH and Ai GX. Satellite selection with multi-objective genetic algorithm for multi-GNSS constellations. *J Shanghai Jiaotong Univ* 2017; 51(12): 1520–1528.
13. Meng X, Nie P, Sun J, et al. A novel satellite selection method based on genetic algorithm. In: Proceedings of the 2018 11th UK-Europe-China workshop on millimeter waves and terahertz technologies, Hangzhou, China, 5–7 September 2018. New York: IEEE.

14. Blanco-Delgado N, Nunes FD and Seco-Granados G. Relation between GDOP and the geometry of the satellite constellation. In: Proceedings of the 2011 international conference on localization and GNSS (ICL-GNSS), Tampa, 29–30 June 2011, pp. 175–180. New York: IEEE.

15. Kim HP, Jang JH, Park JP, et al. Satellite selection method according to signal levels of multi-constellation GNSS. In: Proceedings of the 31st international technical meeting of the satellite division of the institute of navigation (ION GNSS + 2018), Miami, FL, 24–28 September 2018. Manassas, VA: The Institute of Navigation.

16. Liu X, Zhang SB, Zhang QZ, et al. A fast satellite selection algorithm with floating high cut-off elevation angle based on ADOP for instantaneous multi-GNSS single-frequency relative positioning. Adv Space Res 2019; 63(3): 1234–1252.

17. Aguilar MEB, Coury DV, Reginatto R, et al. Multi-objective PSO applied to PI control of DFIG wind turbine under electrical fault conditions. Electr Pow Syst Res 2020; 180: 106081.

18. Dan S and El-Sherief H. Navigation satellite selection using neural networks. Neurocomputing 1995; 7: 247–258.

19. Phatak MS. Recursive method for optimum GPS satellite selection. IEEE Trans Aero Aerosp Electron Syst 2001; 37: 751–754.

20. Gong DW, Sun J and Miao Z. A set-based genetic algorithm for multi-satellite and navigation integrated system. Acta Electron 2009; 37(9): 1931–1936.

21. Li F, Li Z, Gao J, et al. A fast rotating partition satellite selection algorithm based on equal distribution of sky. J Navigation 2019; 72(4): 1053–1069.

22. Hu CY, Pei BN and Cheng C. A satellite selection algorithm for GPS/BDS dual-mode navigation system. Electron Optic Contr 2019; 26(6): 49–53.

23. Gerbeth D, Caamano M, Circiu MS, et al. Satellite selection in the context of an operational GBAS. Navigation 2019; 66(1): 227–238.

24. Xu XJ, Ma LH and Ai GX. Multi-objective and fast satellite selection method based on the NSGA-II algorithm. Astron Res Technol 2018; 15(3): 292–301.

25. Zhou BY. Research on nonlinear predictive control rolling optimization strategy based on SAPSO. In: Proceedings of the IEEE 7th international conference on computational intelligence and security, Sanya, China, 3–4 December 2011. New York: IEEE.

26. Singh P. A novel hybrid time series forecasting model based on neutrosophic-PSO approach. Int J Mach Learn Cyb 2020; 11(8): 1643–1658.

27. Wang M and Li XW. Power cable fault recognition using the improved PSO-SVM algorithm. Appl Mech Mater 2013; 427–429: 830–833.

28. Wang E, Jia C, Tong G, et al. Fault detection and isolation in GPS receiver autonomous integrity monitoring based on chaos particle swarm optimization-particle filter algorithm. Adv Space Res 2018; 61(5): 1260–1272.

29. Tao J, Sun G, Wang XY, et al. Robust optimization for a wing at drag divergence Mach number based on an improved PSO algorithm. Aerosp Sci Technol 2019; 92: 653–667.

30. Kennedy J and Eberhart RC. Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks, Perth, WA, Australia, 27 November–1 December 1995. New York: IEEE.

31. Kuru L, Ozturk A, Kuru E, et al. Determination of voltage stability boundary values in electrical power systems by using the Chaotic Particle Swarm Optimization algorithm. Int J Elect Power Energy Syst 2015; 64(15): 873–879.

32. Wu GH, Wang HL, Pedrycz W, et al. Satellite observation scheduling with a novel adaptive simulated annealing algorithm and a dynamic task clustering strategy. Comput Ind Eng 2017; 113: 576–588.

33. Eberhart RC and Shi YH. Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 congress on evolutionary computation, Seoul, South Korea, 27–30 May 2001. New York: IEEE.

34. Assad A and Deep K. A Hybrid Harmony search and Simulated Annealing algorithm for continuous optimization. Inform Sciences 2018; 450: 246–266.

35. Sudibyo S, Murat MN and Aziz N. Simulated annealing-Particle Swarm Optimization (SA-PSO): particle distribution study and application in Neural Wiener-based NMPC. In: Proceedings of the 2015 10th IEEE Asian control conference (ASCC), Kota Kinabalu, Malaysia, 31 May–3 June 2015, vol. 13, no. 4, pp. 1–6. New York: IEEE.

36. Tang B, Zhu Z, Shin HS, et al. A framework for multi-objective optimisation based on a new self-adaptive particle swarm optimisation algorithm. Inform Sciences 2017; 420: 364–385.

37. Zhang J, Tan Z and Wei Y. An adaptive hybrid model for short term electricity price forecasting. Appl Energ 2020; 258: 114087.

38. Ren M, Huang X, Zhu X, et al. Optimized PSO algorithm based on the simplicial algorithm of fixed point theory. Appl Intell 2020; 50(7): 2009–2024.