

Article

An Improved PSO Algorithm and Its Application in GNSS Ambiguity Resolution

Xin Li ^{1,*} , Jiming Guo ² and Jiyuan Hu ²¹ College of Geology Engineering and Geomatics, Chang'an University, Xi'an 710054, China² School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China; jmguo@sgg.whu.edu.cn (J.G.); plgk@whu.edu.cn (J.H.)

* Correspondence: lixin2017@chd.edu.cn; Tel.: +86-158-0713-9150

Received: 3 May 2018; Accepted: 13 June 2018; Published: 16 June 2018



Abstract: In this study, we proposed a novel method for global navigation satellite system (GNSS) ambiguity resolution (AR). The proposed method utilizes an improved particle swarm optimization (IPSO) algorithm to obtain the GNSS integer ambiguity with the double differenced (DD) float resolution and its corresponding covariance matrix. First, we introduced population maturity to the standard PSO (SPSO) algorithm for the adaptive adjustment of inertia weight. Next, to improve the global convergence and robustness of the SPSO algorithm, we adopted population classification and constructed a Gauss mutation for the particle evolution process of the optimal population. Then, we applied the IPSO algorithm in the field of GNSS AR, called IPSO-AR. Finally, we evaluated the performance of the IPSO-AR algorithm under different DD ambiguity float resolutions with various dimensions and precisions. Numerical results showed that compared with the SPSO-AR algorithm, the IPSO-AR algorithm has a superior correct rate, but low efficiency. Under the appropriate parameter settings, the efficiency of the IPSO-AR algorithm is mainly dependent on the dimensions of DD ambiguity, whereas the correct rate of the IPSO-AR algorithm is mainly dependent on the precision of DD ambiguity. The proposed IPSO-AR algorithm has potential applications under the conditions of few visible satellites or constrained baseline length.

Keywords: global navigation satellite system (GNSS); ambiguity resolution (AR); particle swarm optimization (PSO)

1. Introduction

The carrier phase observation of global navigation satellite systems (GNSS), such as the American Global Positioning System (GPS), Russian Global Navigation Satellite System (GLONASS), European Galileo, and Chinese BeiDou Navigation Satellite System (BDS), is characterized by limited noise and high precision, and is usually applied in high-precision positioning. Ambiguity resolution (AR) is crucial for processing GNSS data, given the initial integer ambiguity of GNSS observations. The correct integer ambiguity must be obtained to transform GNSS observations into high-precision receiver-satellite ranging signals [1–3]. Conventional AR methods first obtain the ambiguity float resolution from GNSS observations. Subsequently, the rounding method, confidence region method, or least-squares method is applied to obtain the correct integer ambiguity [4–7]. The least-squares ambiguity decorrelation (LAMBDA) method has been widely applied in the GNSS AR [8–10].

Numerous intelligent optimization algorithms, such as the neural network algorithm, genetic algorithm, and particle swarm optimization (PSO) algorithm, have attracted increasing attention with the rapid development of computer technology over the past few years [11–13]. The PSO algorithm is a novel evolutionary computation technique that is inspired by social behavior simulation [14–16]. Given its strong global search capability, limited adjustment parameters, and fast computational speed,

the PSO algorithm has attracted considerable research interest. The development of novel applications for the PSO algorithm is currently an important research hotspot.

The PSO algorithm is widely used in several fields, such as function optimization or fuzzy control [17–21]. GNSS AR is essentially a procedure for mapping the relationship between the float solution and integer solution. In the absence of a definite mathematical analytical formula for constructing the relationship between the float solution and the integer solution, some integer ambiguity candidates can be selected and considered in the objective function. The candidate that can satisfy the set condition will be identified as the optimal integer solution. Thus, the integer linear optimal programming problem in mathematics is the core principle of GNSS AR. The PSO algorithm can solve this problem. Therefore, the application of the PSO algorithm in GNSS AR is an interesting topic that nevertheless has received limited research attention. A method for GPS attitude determination based on a discrete PSO algorithm was introduced by [22]. This method lacks ambiguity decorrelation and ignores the defects of the PSO algorithm. An improved PSO algorithm for GPS integer AR was introduced by [23]. However, the performance of this method has not been analyzed in detail, and the relationship between its performance and the property of the ambiguity float solution has been ignored.

In this study, we developed a novel method for GNSS AR by using an improved PSO (IPSO) algorithm, namely, improved particle swarm optimization–ambiguity resolution (IPSO–AR). The ambiguity float resolution and corresponding covariance matrix can be estimated from GNSS observations. Then, we used the PSO algorithm to search for the optimal integer solution by applying a decorrelating Z-transformation [24,25] that is based on the minimum criterion of the weighted residual sum of squares in the corresponding search space. The standard PSO (SPSO) algorithm exhibits some defects, such as proneness to prematurity and to falling into a local optimum. In fact, these defects are common problems of all intelligent optimization algorithms. Thus, the PSO algorithm must be optimized prior to its application. Many previous researchers, such as [26–28], have focused on the improvement of the PSO algorithm.

We introduced population maturity for the adaptive adjustment of inertia weight. In addition, in accordance with the fitness function value, we divided the population into three groups, namely, optimal, suboptimal, and poor populations. Then, we constructed a Gauss mutation to identify the optimal population in particle evolution. The IPSO algorithm can avoid the defects of the SPSO algorithm and exhibits superior performance. We experimentally analyzed the efficiency and correct rate of the IPSO–AR algorithm by using several ambiguity float resolutions with various dimensions and precision under different parameter settings. We identified the factors that affect the performance of the IPSO–AR algorithm, and recommended the optimal parameter settings for the IPSO–AR algorithm on the basis of corresponding characteristics. Then, we used one known baseline to further validate the feasibility of the IPSO–AR algorithm. Finally, we presented some related conclusions.

2. Materials and Methods

The mathematical model of GNSS observations for relative positioning was first presented to obtain the GNSS ambiguity float resolution. Then, an IPSO algorithm was proposed and applied in GNSS AR. This section will focus on IPSO–AR, and its specific application will be illustrated by using a flow diagram.

2.1. Mathematical Model of GNSS Positioning

The positioning observations of GNSS mainly include the pseudorange and carrier phase, denoted as P_r^s and ϕ_r^s , respectively, in the receiver-satellite pair $r - s$. The equations of GNSS observations [29,30] can be formulated as

$$\begin{aligned} P_r^s &= \rho_r^s + T_r^s + I_r^s + c(dt_r - dt^s) + e_r^s \\ \phi_r^s &= \rho_r^s + T_r^s - I_r^s + c(dt_r - dt^s) + \lambda N_r^s + \varepsilon_r^s \end{aligned} \quad (1)$$

where ρ_r^s is the satellite–receiver range; T_r^s and I_r^s are the troposphere delay and ionosphere delay, respectively; dt_r is the receiver clock error, dt^s is the satellite clock error; N_r^s is the carrier phase ambiguity; c is the speed of light; e_r^s and ε_r^s are the remaining error terms; and λ is the wavelength of the corresponding carrier phase.

Through Equation (1), we find that the pseudorange and carrier phase observation have similar structures. Moreover, the latter contains an ambiguity term. This finding implies that as long as ambiguity is resolved correctly, the carrier phase observation will be transformed into the high-precision receiver–satellite pseudorange.

Equation (1) includes two clock bias terms. The classical double differenced (DD) model is usually adopted to eliminate these biases. DD observations are based on simultaneous observations from two receivers and two satellites. These observations can be described as follows:

$$\begin{aligned} P_{qr}^{ls} &= \rho_{qr}^{ls} + T_{qr}^{ls} + I_{qr}^{ls} + e_{qr}^{ls} \\ \phi_{qr}^{ls} &= \rho_{qr}^{ls} + T_{qr}^{ls} - I_{qr}^{ls} + \lambda N_{qr}^{ls} + \varepsilon_{qr}^{ls} \end{aligned} \quad (2)$$

where $\rho_{qr}^{ls} = \rho_r^s - \rho_q^l - (\rho_q^s - \rho_q^l)$, and other terms are similar with ρ_{qr}^{ls} . T_{qr}^{ls} and I_{qr}^{ls} in Equation (2) can be ignored if the baseline length is relatively short (usually less than 10 km) [31,32]. The satellite–receiver range term (ρ_{qr}^{ls}) includes the coordinate parameters of satellites and receivers. The former can be obtained from known ephemeris. ρ_{qr}^{ls} can be linearized with respect to unknown receiver coordinates, represented as the baseline vector in the DD model. The estimated parameters in Equation (2) include three baseline vectors and several ambiguity parameters. The parameter resolution model is an example of a mixed integer-linearized model.

The GNSS DD model can be described as follows:

$$y = Aa + Bb + e, Q_{yy} \quad (3)$$

where y denotes the m -dimensional pseudorange and carrier phase observations; b denotes the three-dimensional baseline vector; a denotes n -dimensional real-valued ambiguity parameters; e denotes the m -dimensional residual errors of observations; A denotes the design matrix that corresponds to a ; B denotes the design matrix that corresponds to b ; Q_{yy} denotes the variance–covariance (VC) matrix that corresponds to the DD pseudorange and carrier phase observation vectors.

We can obtain the solutions of a and b through the least-squares (LS) estimation method, which mainly consists of three steps. We ignore the integrate property of ambiguity parameters in the first step, and use the standard LS method to obtain the real-valued baseline vector and ambiguity parameters.

$$\begin{aligned} \begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} &= \begin{bmatrix} \hat{Q}_{\hat{a}\hat{a}} & \hat{Q}_{\hat{a}\hat{b}} \\ \hat{Q}_{\hat{b}\hat{a}} & \hat{Q}_{\hat{b}\hat{b}} \end{bmatrix} \begin{bmatrix} A^T \\ B^T \end{bmatrix} Q_{yy}^{-1} y \\ Cov \begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} &= \begin{bmatrix} \hat{Q}_{\hat{a}\hat{a}} & \hat{Q}_{\hat{a}\hat{b}} \\ \hat{Q}_{\hat{b}\hat{a}} & \hat{Q}_{\hat{b}\hat{b}} \end{bmatrix} \end{aligned} \quad (4)$$

where \hat{a} and \hat{b} denote the ambiguity and baseline vector parameters of the real-valued resolution, respectively; and $\hat{Q}_{\hat{a}\hat{a}}$ and $\hat{Q}_{\hat{b}\hat{b}}$ are the corresponding VC matrix of \hat{a} and \hat{b} , respectively.

In the second step, we use the estimated float AR to compute the corresponding integer resolution. Numerous methods, such as the popular LAMBDA algorithm, can be adopted in this step. The Z-transformation is usually conducted for ambiguity decorrelation given the strong correlation between ambiguity parameters [24,25].

$$\begin{aligned} \hat{Z} &= T\hat{a} \\ Q_{\hat{z}\hat{z}} &= TQ_{\hat{a}\hat{a}}T^T \end{aligned} \quad (5)$$

where T denotes the nonsingular transformation matrix; \hat{Z} is the ambiguity parameter vectors after decorrelation; and $Q_{\hat{z}\hat{z}}$ is the corresponding VC matrix with Z-transformation for $Q_{\hat{a}\hat{a}}$.

Then, we can search for the optimal integer ambiguity parameters (\bar{Z}) by applying a certain method, such as LS or PSO. The search criteria can be described as

$$f(\bar{Z}) = (\hat{Z} - \bar{Z})^T Q_{\hat{z}\hat{z}}^{-1} (\hat{Z} - \bar{Z}) = \min. \tag{6}$$

Through inverse transformation, we can obtain the fixed ambiguity parameter resolution.

$$\bar{a} = T^{-1}\bar{Z}. \tag{7}$$

In the third step, the remaining real-valued parameters solved in the first step are updated by using the fixed integer parameters:

$$\bar{b} = \hat{b} - Q_{\hat{b}\hat{a}} Q_{\hat{a}\hat{a}} (\hat{a} - \bar{a}). \tag{8}$$

2.2. IPSO-AR Method

2.2.1. Standard PSO Algorithm

Particle swarm optimization is a computational method that optimizes a problem by iteratively attempting to improve a candidate solution in reference to a given fitness function. The SPSO algorithm was developed by introducing the concept of inertia weight into PSO yields [33,34]. With an N -dimensional search space, the total population size is n , assumed as $x_i = (x_{i1} \ x_{i2} \ \dots \ x_{iN})$, $u_i = (v_{i1} \ v_{i2} \ \dots \ v_{iN})$, which can be denoted as the position vector and velocity vector of the particle i . The current optimal position is denoted as $pBest_i$, and the current optimal position of the population is denoted as $gBest$. The SPSO algorithm can be described as

$$\begin{aligned} v_{in}^{(k+1)} &= \omega v_{in}^{(k)} + c_1 r_1 (pBest_{in}^{(k)} - x_{in}^{(k)}) + c_2 r_2 (gBest_{in}^{(k)} - x_{in}^{(k)}), \\ x_{in}^{(k+1)} &= x_{in}^{(k)} + v_{in}^{(k+1)} \end{aligned}, \tag{9}$$

where $i = 1, 2, \dots, n$; ω denotes the inertia weight; c_1 and c_2 are two positive constants, called cognitive learning rate and social learning rate, respectively; and r_1 and r_2 are the random function in the range $[0,1]$.

In this study, different particle positions represent the corresponding integer ambiguity candidate. Their fitness function can be described as

$$F = const - \log[f(\bar{Z})], \tag{10}$$

where F denotes the fitness function value, and $const$ denotes one constant. In this work, $const$ is set as 100.

2.2.2. IPSO Algorithm

Inertia weight is important for the performance of the PSO algorithm and balances the global exploration and local exploitation abilities of the swarm. In the early evolution stage, the value of the particle velocity is expected to be large to enhance global exploration ability. Meanwhile, given that the strong local exploration ability of the PSO algorithm is desired during the late evolution stage, the velocity value is expected to be small [35].

In this study, we introduce the concept of population maturity [36] for the adaptive adjustment of inertia weight on the assumption that $X_{id} = [x_{id} \ p_{id}]$. Population maturity can be described as

$$m_k = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^n \left(1 - \frac{1}{2N} \sum_{d=1}^{2N} |x_{id} - x_{jd}|\right), \tag{11}$$

$$w = \frac{(1-m_0)w_1w_2}{w_2(1-m_k)+w_1(m_k-m_0)}$$

where w_1 and w_2 denote the adjustment range of inertia weight, and m_0 is the initial population maturity. During the particle evolution stage, particle distribution becomes increasingly concentrated. This behavior implies that population maturity continuously increases, whereas the corresponding inertia weight continuously decreases to a certain value.

Although the SPSO algorithm has fast convergence speed, it can easily and rapidly decrease population diversity. Thus, the SPSO algorithm is prone to prematurity and to falling into a local optimum. To avoid these defects, we divide the population into three groups, namely, optimal, suboptimal, and poor populations, in accordance with the fitness function value. These populations are expressed as (S_1, S_2, S_3) , and their corresponding population size is s . We construct a Gauss mutation [37] for the optimal population (S_1) during particle evolution.

$$x_{id}^{k+1} = x_{id}^k + Gauss(0, 1) \cdot x_{id}^k, \tag{12}$$

where $d = 1, 2, \dots, N$; and $Gauss(0, 1)$ denotes the random number that follows Gauss distribution.

We can calculate the average fitness value (F_{ave}) of the optimal population (S_1). The condition for the termination of the IPSO algorithm can be described as

$$F_{max} - F_{ave} \leq afv_out, \tag{13}$$

where F_{max} denotes the individual fitness value of the optimal particle, and afv_out denotes the iterative convergence threshold.

2.2.3. Illustration of the IPSO–AR Algorithm

In accordance with the ambiguity float resolution (\hat{a}) and the corresponding VC matrices ($\hat{Q}_{\hat{a}\hat{a}}$), we can determine the ambiguity search range by using the three times error of estimated \hat{a} or by directly entering one fixed constant into the restricted ambiguity search range.

$$|SearchRange_d| < 3\hat{Q}_{\hat{a}\hat{a}}^{dd} \quad \text{or} \quad |SearchRange| < ar_range, \tag{14}$$

where $1 \leq d \leq N$; $SearchRange$ denotes the ambiguity search range; $\hat{Q}_{\hat{a}\hat{a}}^{dd}$ denotes the estimated error of the d -th dimension ambiguity; and ar_range denotes one given constant.

On the basis of the algorithm described above, we illustrate the specific process of the IPSO–AR algorithm showed in Figure 1.

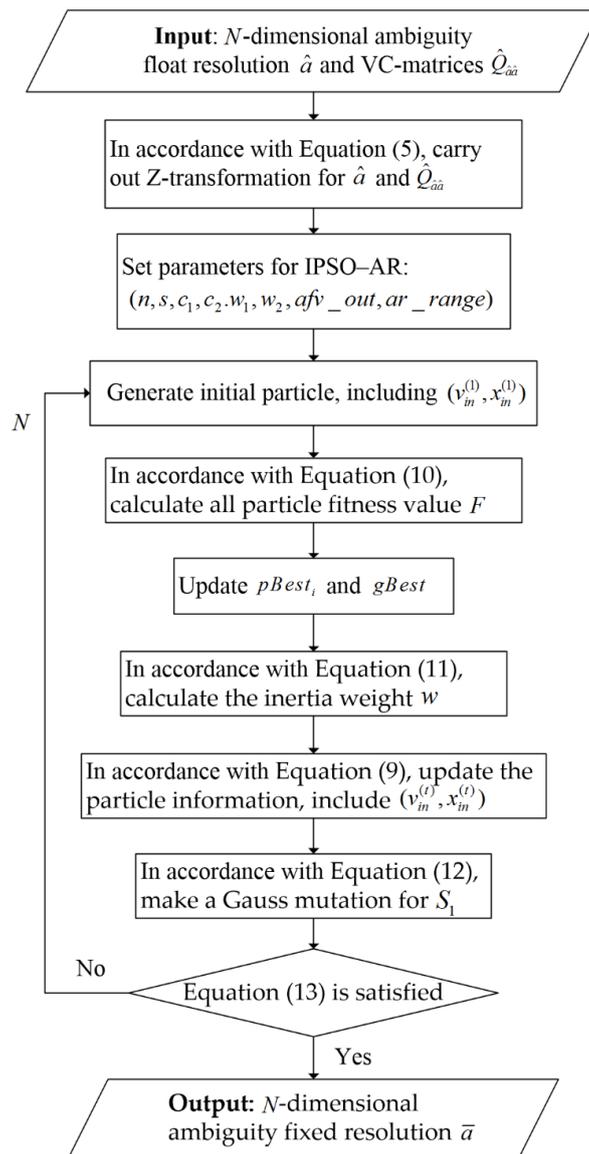


Figure 1. Flow diagram of IPSO-AR (improved particle swarm optimization-ambiguity resolution). VC: variance-covariance matrix.

3. Experiments and Result Analysis

We conducted several experiments to verify the feasibility of the proposed IPSO-AR method. We first demonstrated the search procedure of IPSO-AR. Then, we evaluated the performance of IPSO-AR through three different experiment schemes. We analyzed the influences of the property of the ambiguity float solution and parameter setting on the performance of IPSO-AR in detail. Finally, we used a known baseline to further validate the feasibility of IPSO-AR.

3.1. Search Procedure of IPSO-AR

We select one BDS six-dimensional DD ambiguity float resolution and its corresponding VC matrices as the input information to illustrate the search procedure of IPSO-AR as introduced in Section 2.2.3. Here, we use the ambiguity dilution of precision (ADOP) to assess the precision of ambiguity float resolution. The specific definition and calculation method of ADOP are provided by [38]. In general, a small ADOP value corresponds to a high precision float resolution. In this test,

the ADOP of the ambiguity float resolution is 0.28. The IPSO–AR parameter settings are (90, 30, 0.5, 0.5, 0.4, 0.8, 0.001, 10). Specifically, the total population size is 90. The sizes of the optimal, suboptimal, and poor populations are all 30. The cognitive learning rate and social learning rate are both 0.5. The inertia weight is adjusted over the range of 0.4 to 0.8. The iterative convergence threshold is 0.001. The ambiguity search range is restricted to a finite interval from -10 to 10 .

Figure 2 shows the change trend of the total population maturity and adaptive inertia weight with respect to the number of generations. Notably, the initial value of population maturity is relatively small and is approximately 0.63 because the initial particles are randomly generated with a scattered distribution. Meanwhile, particle distribution becomes increasingly concentrated as the position of individual and global optimum particles is continuously updated. The corresponding population similarity gradually increases. Figure 2 shows that when the number of generations is approximately 10, the value of the total population maturity tends to stabilize (approximately 0.95). The adaptive inertia weight calculated using population maturity also tends to stabilize (approximately 0.5).

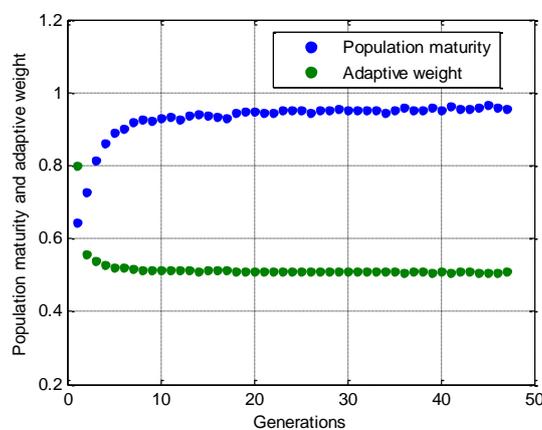


Figure 2. Trend of Population maturity and adaptive weight

Figure 3 shows the evolution of the global optimum. The global optimal fitness completely stabilizes when the number of generations reaches 12. Theoretically, the PSO algorithm has already found the estimated optimal ambiguity integer. However, we utilize several strategies, which are introduced in Section 2.2.2, to avoid falling into a local optimum. The IPSO–AR algorithm satisfies the convergence condition with the strict evolution convergence judgment until the number of generations reaches 47. Thus, the ambiguity search is completed.

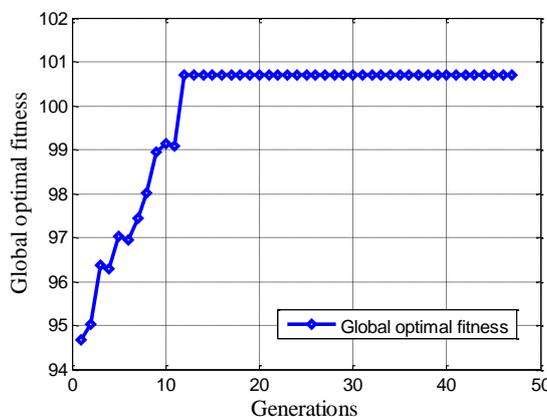


Figure 3. Trend of global optimal fitness.

For a straightforward illustration of the search procedure of IPSO-AR, we select the former second dimension DD ambiguities (i.e., C02-C01 and C07-C01) as an example to represent the evolution of S_1 particles. As shown in Figure 4, the planar distribution of the initial particles, which are shown as solid blue dots (total of 30), is scattered and irregular. Meanwhile, as the number of generations increases, particle evolution became increasingly homogeneous and finally converges to the point (0, -1).

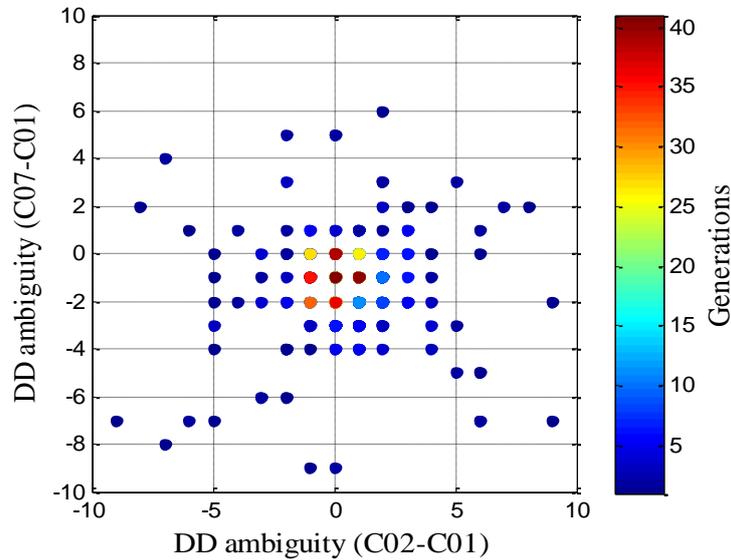


Figure 4. Example of IPSO-AR search procedure. DD: double differenced.

3.2. Performance Analysis of IPSO-AR

In this study, we use one static short baseline (approximately 1.2 km) to validate the performance of IPSO-AR. The average GPS/BDS visible satellite number is 7–8. The truth values of the estimated ambiguity-fixed resolution are obtained from multiepoch process results by using the GNSS open source software RTKLIB [39,40], which users can download from <http://www.rtklib.com>. The processing strategies and methods of RTKLIB software are shown in Table 1.

Table 1. Processing strategies and methods of RTKLIB software. GPS: Global Positioning System; BDS: BeiDou Navigation Satellite System; EKF: extended Kalman filter; LAMBDA: least-squares ambiguity decorrelation.

Main Options	Setting
System	GPS/BDS
Observation Frequencies	L1/B1
Elevation Cut-Off Angle	15°
Positioning Mode	Static
Parameter Estimation	EKF
Stochastic Model	Elevation angle
Satellite Ephemeris	Broadcast
Satellite Antenna Model	IGS08.ATX
Receiver Antenna Model	IGS08.ATX
Ambiguity Resolution	LAMBDA
Ambiguity Validation Threshold	3
Integer Ambiguity Resolution	Continuous
Troposphere Correction	Saastamoinen
Ionosphere Correction	Broadcast Ionosphere Model (Klobuchar Mode)
Earth Tides Correction	OFF
Code/Carrier-Phase Error Ratio	100
Carrier-Phase Error	0.003 + 0.003/sin(el) m

We adopt two indices, namely, efficiency and correct rate, to evaluate the performance of IPSO-AR. Dimension and ADOP can reflect the property of any ambiguity float resolution, which is expected to be related to the performance of IPSO-AR. Meanwhile, similar to the performances of other intelligent search algorithms and as shown by our experimental results, the performance of IPSO-AR depends on parameter settings. Among the nine parameters introduced in Section 2.2.3, the performance of IPSO-AR is the most sensitive to total population size (n) and optimal population size (s).

On the basis of the above analysis, we first consider the influence of ambiguity precision on the performance of IPSO-AR. Thus, we investigate three different experimental schemes that correspond to three different kinds of ambiguity precision (high, general, or low precision) under three different parameter settings (P1, P2, and P3). Only parameters n and s are considered in our experiment. Other parameters are set similarly to those introduced in Section 3.1. The specific details of the experimental schemes are shown in Table 2.

Table 2. Experimental design with various precision double differenced float ambiguities and parameter setting. DD: double differenced; ADOP: ambiguity dilution of precision.

Experiments	DD Float Ambiguity			Parameter Setting		
		Epochs	ADOP	Precision	n	s
Scheme #1	P1	10	(0,0.1]	High	30	10
	P2				60	20
	P3				90	30
Scheme #2	P1	5	(0.1,0.5]	General	30	10
	P2				60	20
	P3				90	30
Scheme #3	P1	1	(0.5,2]	Low	30	10
	P2				60	20
	P3				90	30

We use various dimensions (from 3 to 12) of GPS/BDS DD ambiguity float resolution in independent experiments to analyze the relationship between ambiguity dimension and the performance of IPSO-AR in different schemes with different parameter settings (Table 2). The corresponding satellites and frequency information are provided in Table 3. All the data in our experiment are obtained by the GPS/BDS receiver, and are not simulated.

Table 3. Frequency and satellite information with various DD ambiguity dimensions.

Dimension	$N = 3$	$N = 4$	$N = 5$	$N = 6$	$N = 7$
Frequency	L1	L1	L1	L1	L1 + B1
DD Sat	3G	4G	5G	6G	6G + 1B
Dimension	$N = 8$	$N = 9$	$N = 10$	$N = 11$	$N = 12$
Frequency	L1 + B1	L1 + B1	L1 + B1	L1 + B1	L1 + B1
DD Sat	6G + 2B	6G + 3B	6G + 4B	6G + 5B	6G + 6B

We simultaneously perform experiments on SPSO-AR to compare the performances of IPSO-AR and SPSO-AR. In contrast to IPSO-AR, SPSO-AR uses fixed inertia weight (setting of 0.5). The parameter setting for SPSO-AR is similar to that for IPSO-AR, but lacks the s term.

The computer processor used in our experiment is Intel(R) Pentium(R) CPU G620@2.60 GHz with 4.0 G RAM. All computer programs used standard C language.

3.2.1. Scheme #1 Experiments

In Scheme #1, the selected DD ambiguity float resolutions are obtained from the smoothed results of observation data with 10 epochs. The corresponding ADOP value is less than 0.1, indicating relatively high precision. In IPSO-AR or SPSO-AR, initial particles are generated randomly. We independently conduct the experiment 10,000 times to obtain statistical information (i.e., average efficiency and correct rate).

Figure 5 shows the average number of iterations and search time with various DD ambiguity dimensions for IPSO/SPSO-AR. Under low dimensions (3–5), IPSO-AR and SPSO-AR perform efficiently (approximately 5 ms). However, as the number of dimensions increases, the efficiency of IPSO-AR decreases sharply, but not that of SPSO-AR. When the number of dimensions is 12 under the P3 parameter setting, the average search time of IPSO-AR reaches 1 s, because IPSO-AR constructs a Gauss mutation in the particle evolution of the optimal population (S_1). This condition indicates that some particle evolution paths are disrupted. This disruption, in turn, causes particle evolution to increase and to converge to one optimal position. In addition, the efficiency of IPSO-AR is related to parameter setting. Thus, efficiency is low when total population size (n) is large.

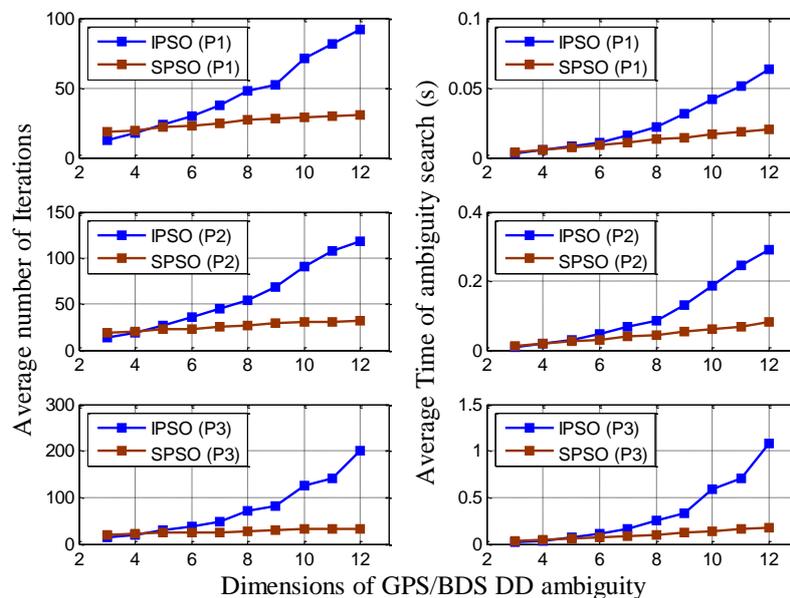


Figure 5. Average number of iterations and search time with various DD ambiguity dimensions for IPSO/standard PSO algorithm (SPSO)-AR under Scheme #1. GPS: Global Positioning System; BDS: BeiDou Navigation Satellite System.

Figure 6 shows the correct rate of IPSO/SPSO-AR under various DD ambiguity dimensions. The correct rate of SPSO-AR decreases sharply under the same parameter setting and increasing dimensions, especially under high dimensions. Meanwhile, the correct rate of IPSO-AR shows the opposite behavior under the same conditions. The correct rate of IPSO-AR/SPSO-AR will increase under a certain dimension of DD ambiguity (Figure 6) and as the parameter n increases. However, the increment in the the correct rate of IPSO-AR is negligible. The correct rate of IPSO-AR is higher than that of SPSO-AR under any condition but is particularly higher under high-dimensional conditions. IPSO-AR can maintain a high correct rate (more than 99.9%) under the following conditions: when the precision of the DD ambiguity float resolutions is relatively high under low-dimensional conditions (3–5), when n is 30 under median-dimensional conditions (6–9), when n is 60 under high-dimensional conditions (10–12), and when n is 90.

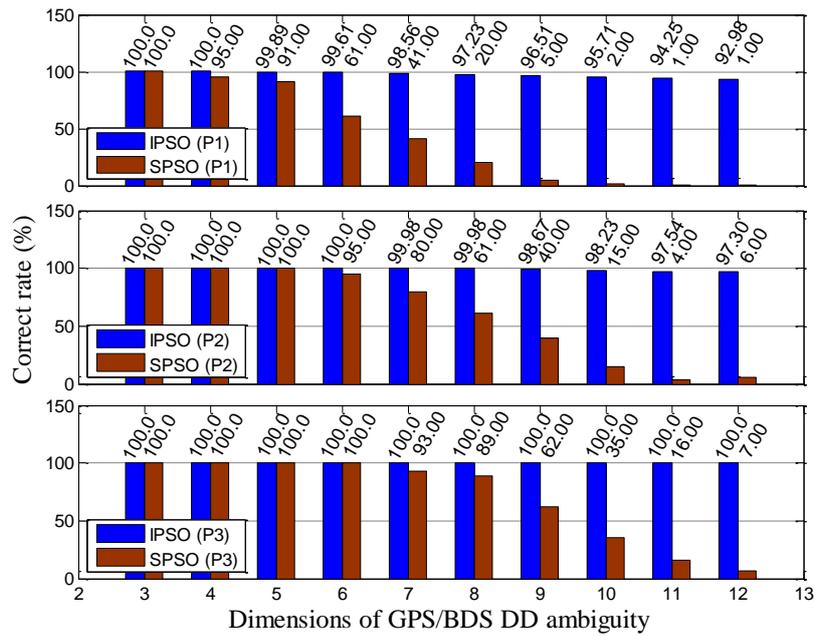


Figure 6. Correct rates of IPSO/SPSO–AR under various DD ambiguity dimensions in Scheme #1.

3.2.2. Scheme #2 Experiments

In Scheme #2, the selected DD ambiguity float resolutions are obtained from the smoothed results of observation data with five epochs. The corresponding ADOP value varies from 0.1 to 0.5. Compared with that obtained under Scheme #1, the precision obtained under Scheme #2 is relatively low. Similar to that in Scheme #1, we conducted the IPSO/SPSO–AR experiment and obtained the average efficiency and correct rate of the algorithms. The results are presented in Figures 7 and 8.

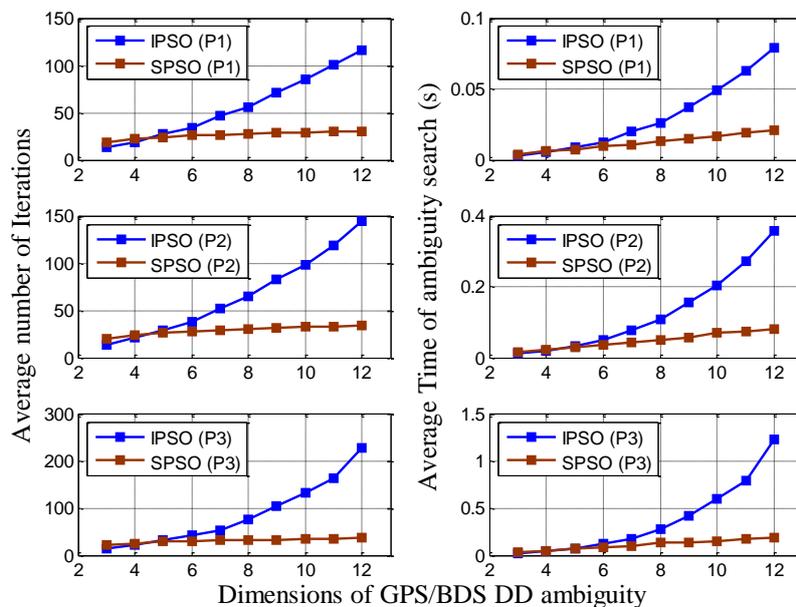


Figure 7. Average number of iterations and search time of IPSO/SPSO–AR under various DD ambiguity dimensions in Scheme #2.

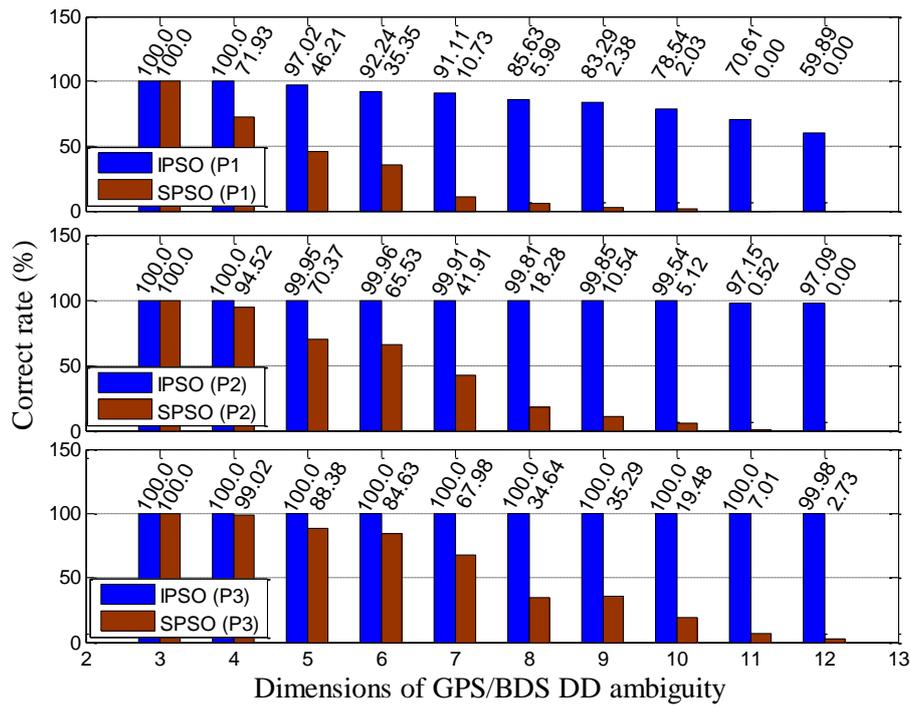


Figure 8. Correct rates of IPSO/SPSO–AR under various DD ambiguity dimensions in Scheme #2.

From Figures 7 and 8, we infer conclusions that are similar to those inferred for Scheme #1. The findings will not be described here to avoid repetition. Comparing Figure 7 with Figure 5, and Figure 8 with Figure 6 shows that under the same parameter setting and ambiguity dimensions, the average efficiency of IPSO/SPSO–AR slightly decreases in Scheme #2 relative to that in Scheme #1. The decrement in efficiency under these conditions, however, is not as drastic as that caused by the increase in dimensions. Under any parameter setting, the correct rate of SPSO–AR in Scheme #2 declines dramatically as the DD ambiguity dimension increases. When the number of dimensions is 10, SPSO–AR completely loses the ability to search for the correct integer ambiguity. Under the P1 parameter setting and high-dimensional conditions, the correct rate of IPSO–AR decreases in Scheme #1 relative to that in Scheme #2. Fortunately, under the appropriate parameter setting, the correct rate of IPSO–AR in Scheme #2 remains high and is similar to that in Scheme #1 (more than 99.9%).

3.2.3. Scheme #3 Experiments

The DD ambiguity float resolutions selected in Scheme #3 are obtained from observations with one epoch. The corresponding ADOP value varies from 0.5 to 2 and indicates the poor precision of the algorithms. The average efficiency and correct rate of IPSO/SPSO–AR in Scheme #3 are shown in Figures 9 and 10.

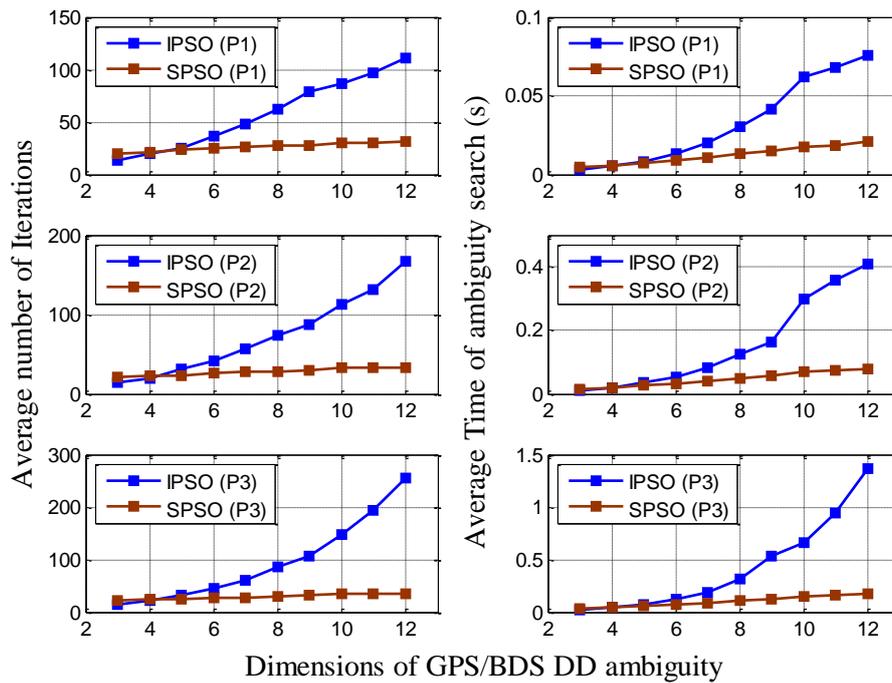


Figure 9. Average number of iterations and search time of IPSO/SPSO–AR under various DD ambiguity dimensions in Scheme #3.

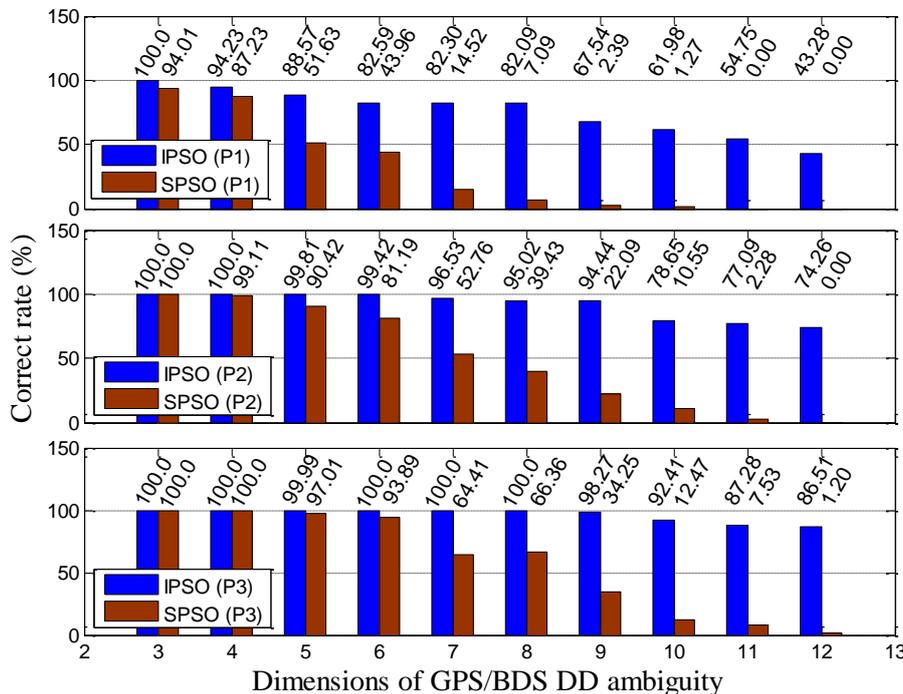


Figure 10. Correct rates of IPSO/SPSO–AR under various DD ambiguity dimensions in Scheme #3.

Comparing Figures 5 and 9, Figures 7 and 10, and Figures 6 and 8, shows that under the same parameter setting, the average efficiency of IPSO/SPSO–AR in Scheme #3 is lower than that in Scheme #1/#2. Although the precision of the DD ambiguity float resolution is low, the correct rate of IPSO–AR in Scheme #3 remains high under the appropriate parameter setting, such as $n = 60$ under low-dimensional conditions, and $n = 90$ under median-dimensional conditions. Meanwhile,

as the number of dimensions continues to increase, the correct rate gradually decreases. Given the IPSO–AR regularities explained in Scheme #1 and Scheme #2, the corresponding correct rate can still be improved even if we increase the value of parameter n . We will not conduct the validation test given the limitations of this study.

3.2.4. Recommended Parameter Settings for IPSO–AR

Through the above analysis, we can find that the performance of IPSO–AR is closely related to the property of DD ambiguity float resolution (dimension and precision). When the dimension reaches 12, the efficiency of IPSO–AR will decrease (more than 1 s). Thus, we will not further discuss our observations for the case of increased dimensions. Meanwhile, low precision (such as ADOP > 3) may result from the inappropriate processing of error sources in the GNSS model or from the inappropriate external observation environment. In this case, searching for the correct integer ambiguity will be difficult, regardless of the AR method. Thus, this case will not be discussed further.

Parameter setting is crucial for the performance of IPSO–AR. We synthetically consider the efficiency and correct rate of IPSO–AR on the basis of the above analysis of the three schemes. Table 4 provides the recommended parameter settings for IPSO–AR under various dimensions and precisions of the DD ambiguity float resolution. Except for parameters n and s , which are presented in Table 4, the other parameters are presented in Section 3.1. We refer to Table 4 to adjust the parameters of IPSO–AR adaptively within the program through the procedure of parameter setting introduced in Section 2.2.3.

Table 4. Recommended parameter setting for IPSO–AR.

ADOP	Dimensions			-
	[1,5]	(5,9]	(9,12]	
(0,0.1]	$n = 30 \ s = 10$	$n = 60 \ s = 20$	$n = 90 \ s = 30$	-
(0.1,0.5]	$n = 30 \ s = 10$	$n = 60 \ s = 20$	$n = 90 \ s = 30$	-
[0.5,2)	$n = 60 \ s = 20$	$n = 90 \ s = 30$	$n = 120 \ s = 40$	-
-	-	-	-	-

3.3. Validation of the IPSO–AR Algorithm with Known Baseline

To further validate the effectiveness of IPSO–AR, we select one baseline with known baseline length and use two GPS receivers. We obtain observations for approximately 2 h at the sampling interval of 1 s and approximately 7000 epochs. The truth value of the baseline length is exactly known with 10.332 m. Then, we postprocess the baseline observation data epoch by epoch by using the real-time kinematic (RTK) model. The proposed IPSO–AR method is used in data processing, and the IPSO–AR parameter setting is illustrated in Table 4.

Figure 11 shows the visible DD satellite number and ADOP value with respect to epochs. Figure 12 shows the assumption time of IPSO–AR with respect to epochs. In most cases, the ADOP value will decrease when the corresponding satellite number increases at some epochs. The redundant observation equation will also increase with the increment in satellite number. Theoretically, the precision of the corresponding DD ambiguity float resolution will improve. Figure 12 shows that at epochs with an increase in satellite number, the corresponding efficiency of IPSO–AR is relatively low. This result proves once again that the efficiency of IPSO–AR is mostly dependent on dimension instead of the precision of DD ambiguity.

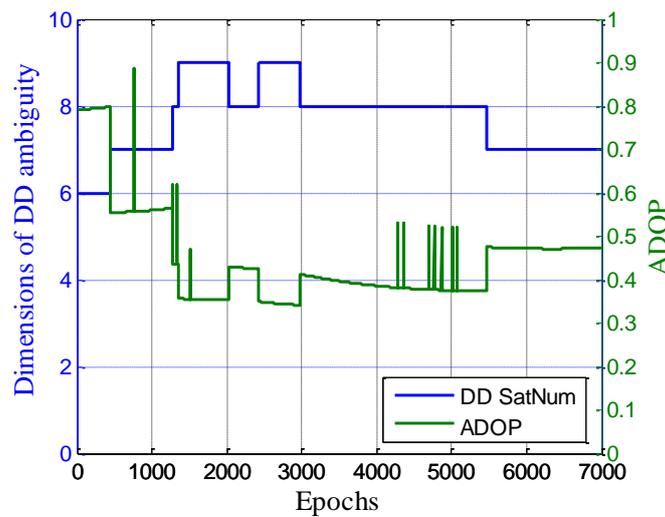


Figure 11. Visible DD satellite number and ADOP value.

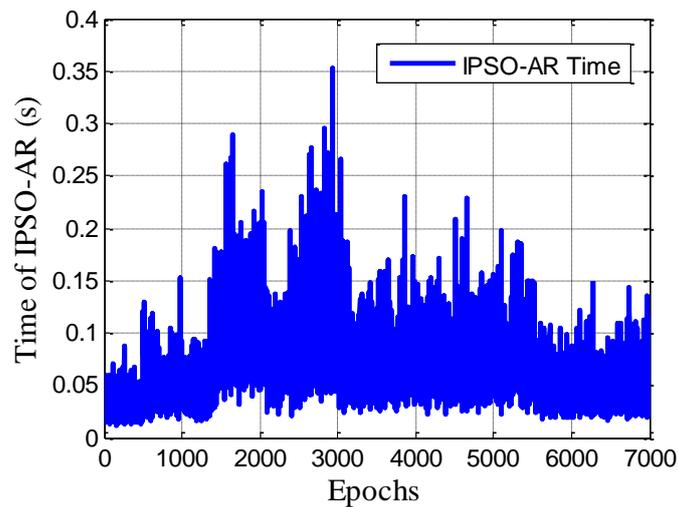


Figure 12. Search time of IPSO-AR.

This experiment verifies the correct rate of IPSO-AR. The series of baseline length differences between the fixed solution (by IPSO-AR method) and truth value is shown as the blue line in Figure 13. An incorrect solution to integer ambiguity may cause large baseline length error. To verify whether the fixed ambiguity calculated through the IPSO-AR method is correct, we randomly add an integer value cycle on one DD satellite at every epoch. The integer value randomly varies from -1 to 1 . The corresponding series of baseline length differences is represented by a green line in Figure 13. Given only one cycle error on the DD ambiguity at many epochs, the corresponding baseline length will deviate from the true value by 10 cm. Meanwhile, all the baseline length errors calculated by IPSO-AR are on the mm level, as represented by the blue line in Figure 13. This result indicates that the estimated DD ambiguity float solutions are all correct. In other words, the correct rate of IPSO-AR in this test is 100%.

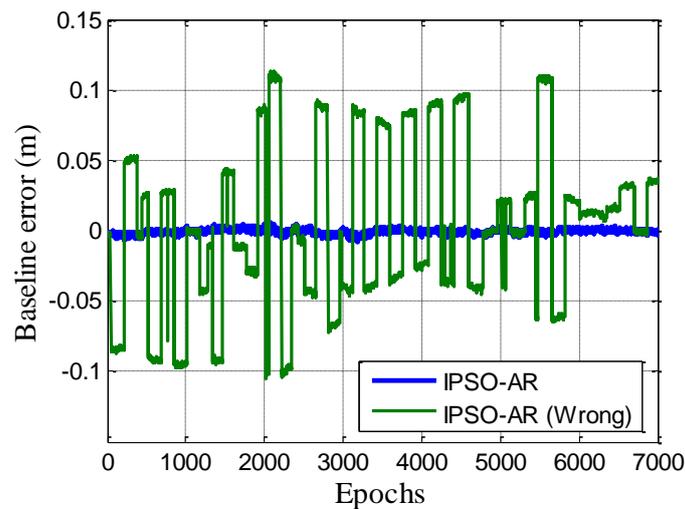


Figure 13. Baseline length errors after single-epoch real-time kinematic (RTK) process with IPSO-AR method.

4. Conclusions

In this study, we firstly proposed an improved PSO algorithm, introduced the population maturity to adjust the inertia weight adaptively, and adopted the strategy of population classification, for the optimal population, we made a Gauss mutation in the process of particle evolution, which can effectively avoid the defect of premature and falling into local optimum existing in SPSO algorithm. Then the proposed IPSO algorithm was applied in the GNSS AR, abbreviated as IPSO-AR. Finally, we conducted a series of experiments with measured data to analyze the performance of IPSO-AR. We obtained the following valuable conclusions:

- (1) The correct rate of IPSO-AR is superior to that of SPSO-AR. The superior correct rate of IPSO-AR, however, is obtained at the expense of computational efficiency. The improvement in the correct rate of IPSO-AR is particularly pronounced under high-dimensional ambiguity.
- (2) The performance of IPSO-AR is closely related to the property of estimated ambiguity float resolution. Under reasonable parameter settings, the efficiency of IPSO-AR mostly depends on the dimension of the estimated ambiguity float resolution. Meanwhile, the correct rate of IPSO-AR mainly depends on the precision of the estimated ambiguity float resolution. Given this principle, we can adaptively adjust the parameters of IPSO-AR on the basis of the estimated ambiguity float resolution, to achieve high efficiency and high correct rate, simultaneously.
- (3) IPSO-AR exhibits high efficiency and high correct rate when the dimension of estimated ambiguity float resolution is low ($N < 6$). The correct rate of IPSO-AR can be validated when baseline length is known, such as in GNSS attitude determination. Thus, IPSO-AR may have considerable engineering application value.

Author Contributions: X.L. devised the research. J.G. and J.H. designed the methods and experiments. X.L. wrote the study. All authors have read and approved the final manuscript.

Funding: This work was supported by the Fundamental Research Funds for the Central Universities, Chang'an University, 2018 (Grant No. 300102268102), the China Postdoctoral Science Foundation, No. 2018M633441, and the Programs of the National Natural Science Foundation of China (41790445, 41774025, and 41731066).

Conflicts of Interest: The authors declare no conflicts of interests.

References

1. Hofmann-Wellenhof, B.; Lichtenegger, H.; Collins, J. *Global Positioning System: Theory and Practice*, 5th ed.; Springer: Berlin/Heidelberg, Germany; New York, NY, USA, 2001.
2. Counselman, C.C.; Gourevitch, S.A. Miniature Interferometer Terminals for Earth Surveying: Ambiguity and multipath with Global Positioning System. *IEEE Trans. Geosci. Remote Sens.* **1981**, *19*, 244–252. [[CrossRef](#)]
3. Blewitt, G. Carrier phase ambiguity resolution for the global positioning system applied to geodetic baselines up to 2000 km. *J. Geophys. Res.* **1998**, *94*, 10187–10203. [[CrossRef](#)]
4. Frei, E.; Beutler, G. Rapid Static Positioning Based on the Fast Ambiguity Resolution Approach “FARA”. Theory and First Results. *Manuscr. Geod.* **1990**, *15*, 325–356.
5. Teunissen, P.J.G. A New Method for Fast Carrier Phase Ambiguity Estimation. In Proceedings of the IEEE Position Location and Navigation Symposium, Las Vegas, NV, USA, 11–15 April 1994.
6. Teunissen, P.J.G. An Optimality Property of the Integer Least-Squares Estimator. *J. Geod.* **1999**, *73*, 587–593. [[CrossRef](#)]
7. Verhagen, S.; Teunissen, P.J.G. New global navigation satellite system ambiguity resolution method compared to existing approaches. *J. Guid. Control Dyn.* **2006**, *29*, 981–991. [[CrossRef](#)]
8. De Jonge, P.J.; Tiberius, C. *The LAMBDA Method for Integer Ambiguity Estimation: Implementation Aspects*; No.12 of LGR-Series, 1996.
9. Verhagen, S.; Li, B.; Teunissen, P.J.G. PS-LAMBDA: Ambiguity correct rate evaluation software for interferometric applications. *Comput. Geosci.* **2013**, *54*, 361–376. [[CrossRef](#)]
10. Chang, X.; Yang, X.; Zhou, T. MLAMBDA: A modified LAMBDA method for integer least-squares estimation. *J. Geod.* **2005**, *79*, 552–565. [[CrossRef](#)]
11. Jiang, G.; Luo, M.; Bai, K.; Chen, S. A Precise Positioning Method for a Puncture Robot Based on a PSO-Optimized BP Neural Network Algorithm. *Appl. Sci.* **2017**, *7*, 969. [[CrossRef](#)]
12. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T.A. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [[CrossRef](#)]
13. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995.
14. Clerc, M.; Kennedy, J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* **2002**, *6*, 58–73. [[CrossRef](#)]
15. Bergh, F.V.D.; Engelbrecht, A.P. A study of particle swarm optimization particle trajectories. *Inf. Sci.* **2006**, *176*, 937–971.
16. Liang, J.J.; Qin, A.K.; Suganthan, P.N.; Baskar, S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans. Evol. Comput.* **2006**, *10*, 281–295. [[CrossRef](#)]
17. Dong, N.; Wu, C.H.; Ip, W.H.; Chen, Z.Q.; Chan, C.Y.; Yung, K.L. An opposition-based chaotic GA/PSO hybrid algorithm and its application in circle detection. *Comput. Math. Appl.* **2012**, *64*, 1886–1902. [[CrossRef](#)]
18. Turgut, O.E.; Turgut, M.S.; Coban, M.T. Chaotic quantum behaved particle swarm optimization algorithm for solving nonlinear system of equations. *Comput. Math. Appl.* **2014**, *68*, 508–530. [[CrossRef](#)]
19. Li, Z.; Tian, Z.; Xie, Y.; Huang, R.; Tan, J. A knowledge-based heuristic particle swarm optimization approach with the adjustment strategy for the weighted circle packing problem. *Comput. Math. Appl.* **2013**, *66*, 1758–1769. [[CrossRef](#)]
20. Zhao, J.; Li, T.; Qian, J. Application of particle swarm optimization algorithm on robust PID controller tuning. In Proceedings of the Advances in Natural Computation, First International Conference, ICNC, Changsha, China, 27–29 August 2005.
21. Mu, B.; Wen, S.; Yuan, S.; Li, H. PPSO: PCA based particle swarm optimization for solving conditional nonlinear optimal perturbation. *Comput. Geosci.* **2015**, *83*, 65–71. [[CrossRef](#)]
22. Xia, N.; Han, D.; Zhang, G.; Jiang, J.; Vu, K. Study on attitude determination based on discrete particle swarm optimization. *Sci. China Technol. Sci.* **2010**, *53*, 3397–3403. [[CrossRef](#)]
23. Li, X.; Zhang, P.; Guo, J.; Wang, J.; Qiu, W. A New Method for Single-Epoch Ambiguity Resolution with Indoor Pseudolite Positioning. *Sensors* **2017**, *17*, 921. [[CrossRef](#)] [[PubMed](#)]
24. Teunissen, P.J.G. The least-squares ambiguity decorrelation adjustment: A method for fast GPS integer ambiguity estimation. *J. Geod.* **1995**, *70*, 65–82. [[CrossRef](#)]

25. Jazaeri, S.; Amiri-Simkooei, A.R.; Sharifi, M.A. Erratum to: Fast integer least-squares estimation for GNSS high-dimensional ambiguity resolution using lattice theory. *J. Geod.* **2012**, *86*, 123–136. [[CrossRef](#)]
26. Tsai, P.W.; Pan, J.S.; Chen, S.M.; Liao, B.Y.; Hao, S.P. Parallel cat swarm optimization. In Proceedings of the 2008 International Conference on Machine Learning and Cybernetics, Kunming, China, 12–15 July 2008.
27. Tsai, P.W.; Pan, J.S.; Chen, S.M.; Liao, B.Y. Enhanced parallel cat swarm optimization based on the Taguchi method. *Expert Syst. Appl.* **2012**, *39*, 6309–6319. [[CrossRef](#)]
28. Chen, S.M.; Chien, C.Y. Parallelized genetic ant colony systems for solving the traveling salesman problem. *Expert Syst. Appl.* **2011**, *38*, 3873–3883. [[CrossRef](#)]
29. Leick, A.; Rapoport, L.; Tatarnikov, D. *GPS Satellite Surveying*, 4th ed.; Wiley: New York, NY, USA, 2015.
30. Li, X.; Guo, J.; Zhou, L. Performance analysis of BDS/GPS kinematic vehicle positioning in various observation conditions. *Sens. Rev.* **2016**, *36*, 249–256. [[CrossRef](#)]
31. Verhagen, S.; Teunissen, P.J.G. Ambiguity resolution performance with GPS and BeiDou for LEO formation flying. *Adv. Space Res.* **2014**, *54*, 830–839. [[CrossRef](#)]
32. Barneveld, P.W.L.V.; Montenbruck, O.; Visser, P.N.A.M. Epochwise prediction of GPS single differenced ionospheric delays of formation flying spacecraft. *Adv. Space Res.* **2009**, *44*, 987–1001. [[CrossRef](#)]
33. Lin, C.; Feng, Q. The Standard Particle Swarm Optimization Algorithm Convergence Analysis and Parameter Selection. In Proceedings of the Third International Conference on Natural Computation (ICNC 2007), Haikou, China, 24–27 August 2007.
34. Bratton, D.; Kennedy, J. Defining a Standard for Particle Swarm Optimization. In Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007), Honolulu, HI, USA, 1–5 April 2007.
35. Pluhacek, M.; Senkerik, R.; Davendra, D.; Oplatkova, Z.K.; Zelinka, I. On the behavior and performance of chaos driven PSO algorithm with inertia weight. *Comput. Math. Appl.* **2013**, *66*, 122–134. [[CrossRef](#)]
36. Zhang, D.X.; Guan, Z.H.; Liu, X.Z. Adaptive particle swarm optimization algorithm with dynamically changing inertia weight. *Control Decis.* **2008**, *23*, 1253–1257.
37. Higashi, N.; Iba, H. Particle swarm optimization with Gaussian mutation. In Proceedings of the IEEE Swarm Intelligence Symposium (SIS 2003), Indianapolis, IN, USA, 26–26 April 2003.
38. Teunissen, P.J.G.; Dennis, O. Ambiguity Dilution of Precision: Definition, Properties and Application. In Proceedings of the 10th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GPS 1997), Kansas City, KS, USA, 16–19 September 1997.
39. Takasu, T.; Yasuda, A. Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB. In Proceedings of the International symposium on GPS/GNSS, Seogwipo-si Jungmun-dong, Korea, 4–6 November 2009.
40. Zhou, F.; Dong, D.; Li, W.; Jiang, X.; Wickert, J.; Schuh, H. GAMP: An open-source software of multi-GNSS precise point positioning using undifferenced and uncombined observations. *GPS Solut.* **2018**, *22*, 22–33. [[CrossRef](#)]

