

## Article

# Multi-Parameter Auto-Tuning Algorithm for Mass Spectrometer Based on Improved Particle Swarm Optimization

Mingzheng Jia <sup>1,2</sup> , Liang Li <sup>3</sup>, Baolin Xiong <sup>1,2</sup>, Le Feng <sup>3</sup>, Wenbo Cheng <sup>2,3,\*</sup> and Wen-Fei Dong <sup>1,2,\*</sup> 

<sup>1</sup> School of Biomedical Engineering (Suzhou), Division of Life Sciences and Medicine, University of Science and Technology of China, Hefei 230026, China

<sup>2</sup> Suzhou Institute of Biomedical Engineering and Technology, Chinese Academy of Sciences, Suzhou 215163, China

<sup>3</sup> Tianjin Key Laboratory of Medical Mass Spectrometry for Accurate Diagnosis, Tianjin 300399, China

\* Correspondence: chengwb@sibet.ac.cn (W.C.); wenfeidong@sibet.ac.cn (W.-F.D.)

**Abstract:** Quadrupole mass spectrometers (QMS) are widely used for clinical diagnosis and chemical analysis. To obtain the best experimental results, mass spectrometers must be calibrated to an ideal setting before use. However, tuning the current QMS is challenging. Traditional tuning techniques possess low automation levels and rely primarily on skilled engineers. Therefore, in this study, we propose an innovative auto-tuning algorithm for QMS based on the improved particle swarm optimization (PSO) algorithm to automatically find the optimal solution of QMS parameters and make the QMS reach the optimal state. The improved PSO algorithm is combined with simulated annealing, multiple inertia weights, dynamic boundaries, and other methods to prevent the traditional PSO algorithm from the issue of a local optimal solution and premature convergence. According to the characteristics of the mass spectrum peaks, a termination function is proposed to simplify the termination conditions of the PSO algorithm and further improve the automation level of the mass spectrometer. The results of auto-calibration testing of resolution and mass axis show that both resolution and mass axis calibration could effectively meet the requirements of mass spectrometry experiments. By the experiment of auto-optimization testing of lens and ion source parameters, these parameters were all in the vicinity of the optimal solution, which achieved the expected performance. Through numerous experiments, the reproducibility of the algorithm was established as meeting the auto-tuning function of the QMS. The proposed method can automatically tune the mass spectrometer from its non-optimal condition to the optimal one, which can effectively reduce the tuning difficulty of QMS.

**Keywords:** quadrupole mass spectrometer; auto-tuning algorithm; improved particle swarm optimization algorithm; inertia weights; dynamic boundaries



**Citation:** Jia, M.; Li, L.; Xiong, B.; Feng, L.; Cheng, W.; Dong, W.-F. Multi-Parameter Auto-Tuning Algorithm for Mass Spectrometer Based on Improved Particle Swarm Optimization. *Bioengineering* **2023**, *10*, 1079. <https://doi.org/10.3390/bioengineering10091079>

Academic Editors: Antonio Boccaccio, Luca Mesin and Longjun Liu

Received: 22 July 2023

Revised: 19 August 2023

Accepted: 5 September 2023

Published: 12 September 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The quadrupole mass spectrometer (QMS) is a highly efficient mass spectrometer with excellent sensitivity and selectivity for various analytical applications [1,2], such as clinical diagnosis, the identification and quantification of biomolecules [3], drug metabolism studies [4], and environmental monitoring [5]. With the widespread use of QMS mass spectrometers in recent years, users not only inquire about the higher sensitivity of the instrument, but also pursue the intelligence and automation of the instrument, with analytical experiments performed quickly [6]. Mass spectrometer tuning is the process of optimizing the mass spectrometer parameters to achieve optimal experimental results before performing a sample analysis [7]. The goal of tuning is to obtain reasonable resolution and accurate mass axes, as well as acceptable peak intensities.

However, the mass spectral resolution and spectral peak intensity conflict with each other; that is, a higher resolution tends to degrade the intensity, while a higher intensity

suppresses the resolution. Therefore, a trade-off between resolution and spectral peak intensity must be made in tuning and ensuring that both are met to obtain better experimental results. In addition, there is a degree of correlation between the resolution and mass axis, and the purpose of tuning is to find a balance between them such that the instrument can reach the optimal state.

Mass spectrometer tuning is divided into manual tuning and auto-tuning. In general, manual tuning is difficult and time consuming in addition to having inconsistent standards. Thus, auto-tuning is the current practice in mainstream instruments and is an important function in modern mass spectrometers, which can make mass spectrometry more rapid, accurate, and reliable, and help to improve the level of mass spectrometry automation. Currently, many studies focused on tuning mass spectrometry resolution in terms of automatic tuning of mass spectrometers. By adjusting the direct current (DC) and alternating current (AC) voltages applied to the quadrupole to change the resolution of the mass spectrometer, Kenny [8] proposed a method for adjusting the DC and AC voltages of the quadrupole to change the resolution of the mass spectrometer; however, the scheme design is not mentioned in this study. Syed et al. [9] proposed a ratiometric relationship between the DC and AC voltages and the resolution of the mass spectrometer, but did not realize the function of automatic resolution adjustment. Recently, Liu et al. [10] established an automatic mass resolution tuning algorithm, which equipped a corresponding model for tuning through the linear relationship between mass resolution and electrical parameters. However, the model only tuned the resolution without tuning the mass axis and spectral peak intensity. Moreover, the method often fails to achieve the optimal state of the instrument because of its use of iterative algorithms that fall into a state of local optimal solutions.

Therefore, how to solve the optimal solution problem in auto-tuning became a hot spot of concern. Automatic tuning methods based on heuristic optimization algorithms are more suitable for multi-parameter optimization, such as genetic algorithms, particle swarm optimization (PSO), and so on. The PSO algorithm is a heuristic optimization algorithm based on group intelligence, which has the advantages of simple implementation, fast convergence, and easy parallel computation, and is mostly used to solve combinatorial optimization, mode decomposition, sensor networks, biomolecular research, and other fields [11]. However, the traditional PSO algorithm has some limitations in optimization problems with multi-peak functions, such as poor convergence of the population of particles and poor global search ability of the algorithm [12–16]. To improve the performance of PSO algorithms, there were many studies on improved PSO algorithms. J. Riget introduced the attractive and repulsive PSO (ARPSO) in trying to overcome the problem of premature convergence, which uses a diversity measure to control the swarm [12]. Yu et al. proposed an enhanced PSO algorithm to enhance the convergence rate of iteration process [13]. Yu et al. proposed a nonparametric model for a magnetorheological elastomer-based isolator based on support vector regression (SVR), which employs a type of improved particle swarm optimization to optimize the parameters in SVR and has a better generalization capacity and better recognition accuracy than other conventional models [14]. F. Bergh proposed the cooperative particle swarm optimizer by using multiple swarms to optimize different components of the solution vector cooperatively to improve the performance of the traditional PSO algorithm [15]. Yue et al. proposed a PSO-BOA optimization strategy with applications in data classification [16]. The improvement of the traditional PSO algorithm is mainly divided into multiple populations to search; using topological structure to improve; combining with other evolutionary algorithms; improvement and change in iterative formula of velocity and position, etc., which enhance the convergence rate and search ability [15–19].

In this paper, we propose an improved particle swarm optimization (I-PSO) algorithm. This method introduces a simulated annealing (SA) algorithm, boundary dynamics, and improved multi-inertia weight selection, which improve the algorithm convergence rate and prevent the particles from falling into the local optimum. Additionally, we propose a QMS auto-tuning method based on an improved particle swarm optimization (I-PSO)

algorithm. The I-PSO algorithm combined with simulated annealing and the multi-inertia weight algorithm is used to auto-tune the QMS instrument and optimize the quality of the spectra, which can make the QMS instrument reach the optimal state.

## 2. Methods

### 2.1. Standard PSO Algorithm and Simulated Annealing Algorithm

The PSO algorithm consists of  $M$  particles, each representing a potential solution. When the particles move, they consider their best position in the history and the group's overall best position to gradually adjust their direction and speed toward the optimal solution [20].

The equation for particle  $j$  to adjust the velocity is shown in Equation (1).

$$v_j(n + 1) = wv_j(n) + c_1(p_j(n) - x_j(n)) + c_2(g_j(n) - x_j(n)), j = 1 \cdots M \quad (1)$$

Here,  $w$  is the inertia weight, and constants  $c_1$  and  $c_2$  are random numbers,  $p_j(n)$  is the historical best position of particle  $j$ , and  $g_j(n)$  is the global best position of all particles. Equation (2) expresses the position-updating process for particle  $j$ .

$$x_j(n + 1) = x_j(n) + v_j(n + 1), j = 1 \cdots M \quad (2)$$

Here,  $x_j(n + 1)$  and  $v_j(n + 1)$  represent the position coordinates and velocity vector, respectively, at the time step  $n + 1$ .

The steps of the standard PSO algorithm [21] are as follows:

**Input:** Velocity boundary  $v_{\min}, v_{\max}$ , position boundary  $x_{\min}, x_{\max}$ , maximum number of iterations  $N$ , and initial velocity is set to 0.

**Step 1:** Initialization: Randomly generate the positions of all particles  $x_1(0) \cdots x_M(0)$ , and obtain the global best position  $g(0)$  at the current time. Proceed to Step 2.

**Step 2:** Calculate the velocity of each particle  $x_j(n)$  at  $v_j(n + 1)$  according to Equation (1). If  $v_j < v_{\min}$  then  $v_j = v_{\min}$ ; if  $v_j > v_{\max}$  then  $v_j = v_{\max}$ .

**Step 3:** Update the  $x_j(n)$  position of each particle according to Equation (2),

if  $x_j(n + 1) < x_{\min}$  then  $x_j(n + 1) = x_{\min}$ ,

if  $x_j(n + 1) > x_{\max}$  then  $x_j(n + 1) = x_{\max}$ . Go to Step 4.

**Step 4:** Calculating the global best position,

$$g(n + 1) = \operatorname{argmax}\{f(x_j(i + 1)), j = 1 \cdots M, i = 0 \cdots n\} \quad (3)$$

**Step 5:** Then check if the termination condition is met. If yes, terminate the algorithm. Otherwise, return to Step 2.

**Output:** Global best position  $g(n)$ .

The simulated annealing (SA) algorithm is a versatile global optimization algorithm that works by exploring the solution space through random walks to find the best possible solution [22,23]. Two main processes are involved in the SA algorithm: the Metropolis algorithm and annealing process. The Metropolis algorithm is a randomized method that determines whether to accept a new solution as the current solution to prevent it from becoming stuck in a local optimal solution. The Metropolis algorithm criteria are expressed by Equation (4).

$$P_{ij} = \begin{cases} 1 & f(i) \geq f(j) \\ \exp[\frac{f(i)-f(j)}{T}] & f(i) < f(j) \end{cases} \quad (4)$$

Here,  $T$  is the temperature parameter and  $f(i)$  and  $f(j)$  are the internal energies of the solid in states  $i$  and  $j$ , respectively.

When  $f(i) \geq f(j)$ , the SA algorithm accepts this new state with 100% probability. Therefore, the SA method eventually converges to the global optimal solution [24].

## 2.2. Improved PSO Algorithm

The standard PSO algorithm uses a constant inertial weight. If the inertia weight is low, the algorithm can easily determine the global optimal solution within the initial search space. However, if the optimal solution is outside this space, it may not be found. Nevertheless, if the inertia weight is large, the algorithm behaves more as a global search method and always searches for new areas. However, more iterations are required to reach the global optimal solution. If the inertia weight is moderate, the algorithm has a higher probability of finding the global optimal solution, but requires more iterations than in the first case. To address these issues, researchers proposed a method for random inertia weights. However, this method has high randomness [25–27]. In this study, a multi-inertia weight selection method was proposed based on the random inertia weight method. Thus, several inertia weights were added, several speeds were updated in each iteration, and several new positions were obtained. The best model was selected for comparison.

This was achieved by transforming the constant inertia weights in Equation (1) into a number of inertia weights that vary with the number of iterations. In each iteration, the inertia weights traverse all the elements in the set of inertia weights and find the new velocity corresponding to each element in the set of inertia weights. The set of inertia weights in this study contained four elements that linear weight, exponential weight, power weight and random weight. The mathematical form of the inertia weights are as follows [27–29].

Linear weight(L):

$$w(n) = w_{\max} - \frac{n}{N}(w_{\max} - w_{\min}) \quad (5)$$

Here,  $w_{\max}$  is the max value of the weight,  $w_{\min}$  is the min value of the weight,  $N$  is the iteration max value,  $n$  is the algorithm iterations.

Exponential weight (E):

$$w(n) = w_{\max}e^{-\frac{n}{N}} \quad (6)$$

Here,  $w_{\max}$  is the max value of the weight,  $N$  is the iteration max value,  $n$  is the algorithm iterations.

Power weight (P):

$$w(n) = w_{\max} - (w_{\max} - w_{\min})\left(\frac{n}{N}\right)^2 \quad (7)$$

Here,  $w_{\max}$  is the max value of the weight,  $w_{\min}$  is the min value of the weight,  $N$  is the iteration max value,  $n$  is the algorithm iterations.

Random weight(R):

$$w(n) = rand(w_{\min}, w_{\max}) \quad (8)$$

Here,  $w_{\max}$  is the max value of the weight,  $w_{\min}$  is the min value of the weight,  $n$  is the algorithm iterations.

Another drawback of the standard PSO algorithm is its slow convergence rate, especially in the vicinity of the optimal solution; studies showed that the PSO algorithm tends to enter a stagnant state in the late evolutionary stage [30]. To resolve this problem, a dynamic boundary correction method can be added to the standard PSO algorithm; that is, the position boundary changes with the number of iterations. The advantage of the proposed method is that it improves the operational efficiency of the algorithm by weakening the oscillation amplitude when the algorithm converges, making it easier to satisfy the termination condition when the algorithm evolves to a late stage [31].

The dynamic correction of its boundary is determined by changing the fixed position boundary into a dynamic position boundary in Step 2 of the standard PSO algorithm. In general, the range of the dynamic position boundary is subsequently narrowed for

faster convergence. The dynamic update of the position boundaries is calculated using Equations (9) and (10).

$$x_{\min} = \begin{cases} p_j(n) - \alpha|p_j(n) - x_j(n)| & p_j(n) \neq x_j(n) \\ p_j(n) - \beta|p_j(n)| & p_j(n) = x_j(n) \end{cases} \quad (9)$$

$$x_{\max} = \begin{cases} p_j(n) + \alpha|p_j(n) - x_j(n)| & p_j(n) \neq x_j(n) \\ p_j(n) + \beta|p_j(n)| & p_j(n) = x_j(n) \end{cases} \quad (10)$$

The probability of the particle jumping out of the local optimal solution is increased by enhancing the search capability of the algorithm by adding SA to the above two improvements. The steps of the improved PSO (I-PSO) algorithm with the addition of SA, boundary dynamics, and multiple inertial weights are as follows:

**Input:** Velocity boundary  $v_{\min}, v_{\max}$ , position boundary  $x_{\min}, x_{\max}$ , maximum number of iterations  $N$ , initial velocity is set to 0, parameter  $\alpha, \beta$ , SA parameters  $\varepsilon, \eta$ , and  $T$ .

**Step 1:** Initialize: Randomly generate the positions of all the particles  $x_1(0) \cdots x_M(0)$  and obtain the global best position  $g(0)$  at the current time. Proceed to Step 2.

**Step 2:** Update the position boundary  $x_{\min}, x_{\max}$  dynamically according to dynamic boundary Equations (9) and (10).

**Step 3:** The velocity  $v_j(n + 1)$  is calculated for each particle  $x_j(n)$ , as shown in Equation (7); however, four weights need to be calculated separately.

$$v_j(n + 1) = wv_j(n) + c_1(p_j(n) - x_j(n)) + c_2(g(n) - x_j(n)), j = 1 \cdots M \quad (11)$$

Here,  $w = w_{\Phi}(n)$ ,  $\Phi = \{L, E, P, R\}$  represents the four corresponding inertia weight Equations (5), (6), (7) and (8), respectively. For convenience, the four calculated velocities are denoted as  $v_j^L, v_j^E, v_j^P, v_j^R$ .

If  $v_j^{\Phi} < v_{\min}$  then  $v_j^{\Phi} = v_{\min}$ , else If  $v_j^{\Phi} > v_{\max}$  then  $v_j^{\Phi} = v_{\max}$ .

**Step 4:** Update the position of each particle  $x_j(n)$ , as shown in Equation (2). According to the four velocities obtained in Step 3, four new positions, denoted as  $x_j^L(n + 1)$ ,  $x_j^E(n + 1)$ ,  $x_j^P(n + 1)$ ,  $x_j^R(n + 1)$ , are obtained. Subsequently, the optimal solutions for the four new positions were calculated.

$$x_j(n + 1) = \operatorname{argmax}\{f(x_j^{\Phi}(n + 1)), \Phi = \{L, E, P, R\}\} \quad (12)$$

If  $x_j(n + 1) < x_{\min}$  then  $x_j(n + 1) = x_{\min}$ , else if  $x_j(n + 1) > x_{\max}$  then  $x_j(n + 1) = x_{\max}$ . Go to Step 5.

**Step 5:** Calculate the global optimal position and particle history optimal position, which are updated using the SA algorithm.

$$\Delta E_j = f(p_j(n)) - f(x_j(n + 1))$$

$$\Delta E = f(g(n)) - \max\{f(x_j(n + 1)), j = 1 \cdots M\} \quad (13)$$

Based on this there are:

$$p_j(n + 1) = \begin{cases} \operatorname{argmax}\{f(x_j(i + 1)), i = 0 \cdots n\} & e^{-\frac{\Delta E}{T}} > \varepsilon \\ f_j(x_j(n + 1)) & e^{-\frac{\Delta E}{T}} \leq \varepsilon \end{cases} \quad (14)$$

$$g(n + 1) = \begin{cases} \operatorname{argmax}\{f(x_j(i + 1)), j = 1 \cdots M, i = 0 \cdots n\} & e^{-\frac{\Delta E}{T}} > \varepsilon \\ \operatorname{argmax}\{f(x_j(n + 1)), j = 1 \cdots M\} & e^{-\frac{\Delta E}{T}} \leq \varepsilon \end{cases} \quad (15)$$

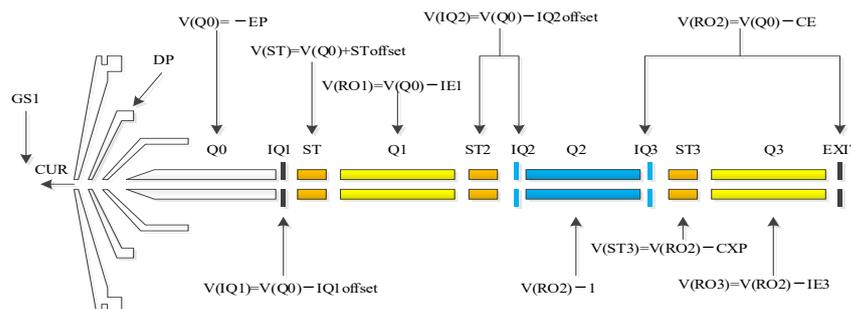
$$T = \eta T, \eta < 1$$

Here,  $\varepsilon$  is a random number generated in a uniform distribution of  $[0, 1]$ .

If the termination condition is reached, reach the end; otherwise, return to **Step 2**.  
**Output:** Global best position  $g(n)$ .

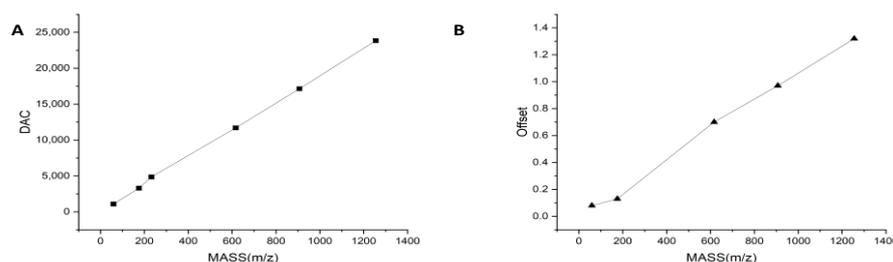
### 2.3. Auto-Tuning Algorithm for QMS Based on Improved PSO Algorithm

The tuning of QMS aims to make the mass spectrometer work optimally, which requires mass axis parameter calibration, resolution parameter (also known as the full width at half maximum (FWHM)) adjustment, and optimization of lens and ion source parameters for the mass spectrometer. The most commonly used QMS is the triple QMS, the ion optical schematic of which is shown in Figure 1.



**Figure 1.** Schematic diagram of the lens and ion source parameters of a triple quadrupole mass spectrometer.

The mass axis of a QMS is determined by the linear proportionality between several mass numbers and digital-to-analog converters (DACs). For convenience, we formed pairs of mass numbers, DACs, and defined them as  $(\text{MASS}_1, \text{DAC}_1) \dots (\text{MASS}_n, \text{DAC}_n)$ . The MAS-DAC curve can be adjusted by appropriately adjusting the DAC values. For example, if DAC3 is adjusted, the linear relationships between  $(\text{MASS}_2, \text{DAC}_2)$  and  $(\text{MASS}_3, \text{DAC}_3)$ ,  $(\text{MASS}_3, \text{DAC}_3)$  and  $(\text{MASS}_4, \text{DAC}_4)$  change. Similarly, the resolution is controlled by several MASS-OFFSET pairs, and the relative MASS-OFFSET curves can be changed by appropriately adjusting the corresponding offset values. The correspondence between the mass and DAC and the correspondence between the mass and offset are shown in Figure 2.



**Figure 2.** Correspondence of mass to DAC and mass to offset; (A) correspondence of mass to DAC; and (B) correspondence of mass to offset.

The lens parameters are a series of DC voltages acting on charged particles to obtain more ions from the mass analyzer and detector. To achieve the highest transfer efficiency, ion source parameters, which determine the ionization efficiency of the ion source, are also important for mass spectrometry. With proper settings of the lens and ion source parameters, the efficiency of material ionization and transport to the detector is significantly increased, and a better response can be obtained. Therefore, the optimization of these two types of parameters is crucial.

Based on the aforementioned characteristics of quadrupole mass spectrometry, we propose a termination condition evaluation function applicable to the improved PSO algorithm, as shown in Equation (16).

$$f(x) = \frac{\lambda_1}{I} + \lambda_2 \|FW - TFW\|^2 + \lambda_3 \|MASS - TMASS\|^2, x = (x_f, x_{len}, x_m)^T \quad (16)$$

Here,  $x_f$ ,  $x_{len}$ , and  $x_m$  are the resolution parameter vector, lens and ion source parameter vector, and mass axis parameter vector, respectively;  $I$  is the mass spectral peak intensity;  $FW$  and  $TFW$  are the current mass spectral peak FWHM and target FWHM of the mass spectral peak, respectively;  $MASS$  and  $TMASS$  are the detected mass numbers of the current mass spectral peak and target mass number of the mass spectral peak, respectively; and  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$  are non-negative constants. This evaluation function is also an important basis for determining whether an instrument is in optimal condition. In general, without any a priori knowledge,  $x_s = \operatorname{argmin} f(x)$  is the optimal value for any input parameter  $x$ , and thus the instrument is in the best condition. For the condition with a priori knowledge, a threshold value  $\varepsilon$  is set. The instrument is in the best condition when  $f(x) \leq \varepsilon$  and the parameter satisfying  $f(x) \leq \varepsilon$  is the optimal parameter.

For a QMS, the tuning procedure encompasses three distinct stages: mass axis calibration, resolution calibration, and parameter optimization. The parameter optimization included lens parameter optimization and ion source parameter optimization. The initial steps involved calibration of the mass axis and resolution, and it is worth noting that these two aspects exhibited a limited correlation. During the execution, iterative alternation between the mass axis and resolution calibration was performed. Subsequently, the parameter optimization process commenced and concluded when the intensity reached a predefined threshold. At this point, the final values of the mass axis, resolution, and other parameters were obtained, thereby completing the tuning process. A flowchart of the QMS auto-tuning algorithm based on an improved PSO algorithm (I-PSO-Tune) is shown in Figure 3.

First, we discuss the mass axis calibration problem, where the mass axis control parameter shows a linear relationship ( $R > 0.99$ ) with the mass number, which we assume to be of the form in Equation (17).

$$DACs = \omega MASS + \zeta \quad (17)$$

Here,  $MASS$  is the mass number,  $DACs$  are the corresponding mass axis control parameter values,  $\omega$  is the slope, and  $\zeta$  is the intercept, which should be zero under ideal conditions. The correction equation for the mass axis is as in (18).

$$DACs(n) = DACs(n - 1) + \omega(MASSd + MASS(n)) \quad (18)$$

Here,  $DACs(n)$  is the DAC value obtained from the  $n$ th iteration, which is the DAC value of the last time of  $DACs(n - 1)$ ,  $\omega$  is the slope in Equation (18),  $MASSd$  is the target mass number, that is, the mass number corresponding to the peak of the tuned liquid spectrum, and  $MASS(n)$  is the mass number corresponding to the peak of the  $n$ th detected spectrum.

Next, we discuss the resolution calibration problem. When adjusting the resolution of  $N$  feature mass numbers simultaneously with whose  $OFFSET$  value presents certain correlations, vectors comprising these  $N$  parameters are input to the improved PSO algorithm model with the output as the new  $OFFSET$  value. Termination conditions for the mass axis and resolution are shown in flowchart (I) as follows: set the parameters in Equation (12) to  $\lambda_1 = 0$ ,  $\lambda_2 > 0$ , and  $\lambda_3 > 0$ .

Finally, we discuss the parameter optimization problem in which we set the termination condition (II) of the flowchart as follows: set the parameters in Equation (16) to  $\lambda_1 = I_T$ ,

$\lambda_2 = 0$ , and  $\lambda_3 = 0$ , where  $I_T$  is the mass spectrum response value. Thus, the iterative stopping condition for parameter optimization is

$$\|f(x(n))\| \leq \varepsilon, x(n) = (x_f, x_{len}, x_m)^T(n) \text{ or } n = N. \tag{19}$$

Here,  $\lambda_1 = I_T$ ,  $\lambda_2 = 0$  and  $\lambda_3 = 0$ , and  $\varepsilon = 1$ .

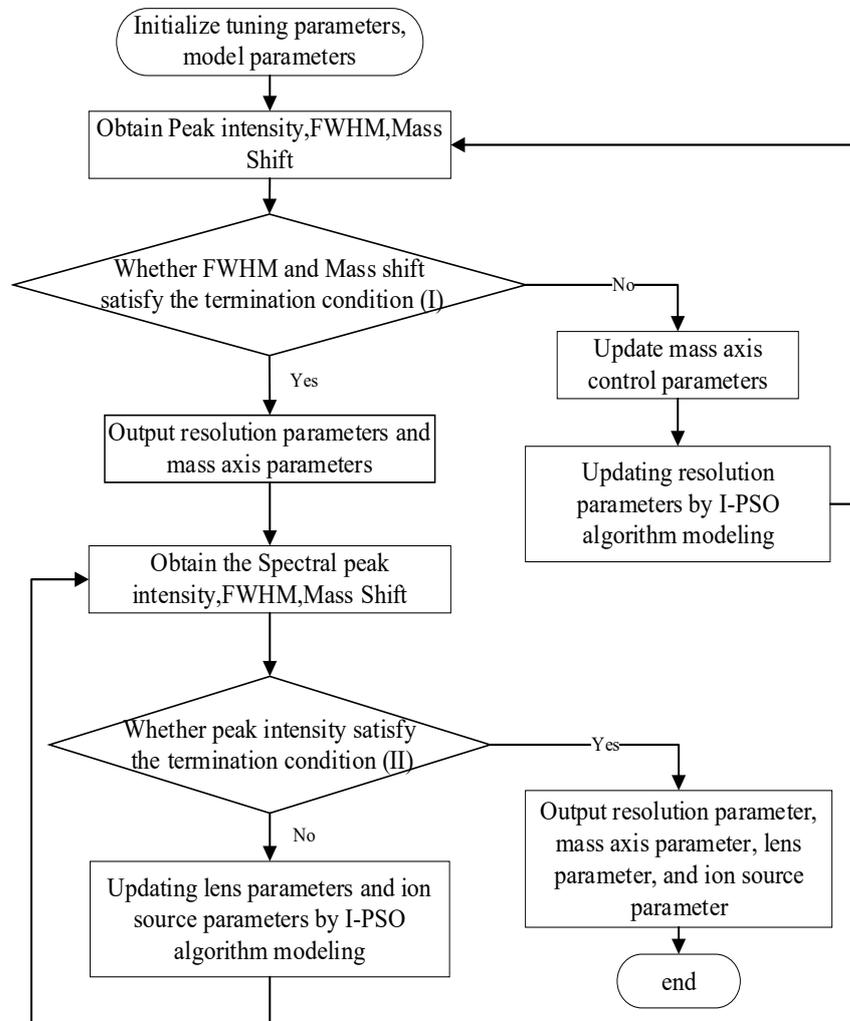


Figure 3. Flow chart of QMS auto-tuning algorithm based on improved PSO algorithm.

### 3. Results and Discussion

#### 3.1. Benchmark Function Test

In order to test the performance of the I-PSO algorithm, we select the standard PSO algorithm and I-PSO algorithm for a benchmark function test. In this, eight functions are selected to test the algorithm performance. All eight test functions in CEC2017 are evaluated as minimization problems and are classified as unimodal functions, simple multimodal functions, hybrid functions, and composite functions [32]. The test functions are shown in Table 1. The solution dimension of all test functions is 10, the population size is set to 100, and the search space is  $[-100, 100]$ , and all algorithms are run independently on each test function 10 times, with the maximum number of iterations for each run being [100, 1500].

Based on the above benchmark functions test results, it can be analyzed that I-PSO makes it easier to jump out of the local optimum and reach the global optimum point compared with the traditional PSO algorithm. The reason for this is that I-PSO adds the multi-inertia weight approach and the simulated annealing algorithm in PSO. The

multi-inertia weight approach can expand the search range of the algorithm to be able to perceive more regions, while the SA algorithm provides the possibility of jumping out of the local optimum, because of the addition of simulated annealing, the I-PSO has a certain probability of rejecting the current optimum point, so as to achieve the purpose of jumping out of the local optimum solution. In the experiment, the function with multiple extreme points is selected, shaped as (E)F5~(H)F8 in Figure 4, and it can be seen that I-PSO can reach its optimal point; however, the traditional PSO will fall into the local optimum.

**Table 1.** Test functions in CEC2017 [32].

Functions Type	Functions No.	Functions	$F_i = F_i(x^*)$
Unimodal function	F1	Shifted and rotated sum of different power function $f_2(x) = \sum_{i=1}^D  x_i ^{i+1}$	200
	F2	Shifted and rotated rosenbrock's function $f_4(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$	400
Simple multimodal functions	F3	Shifted and rotated expanded Scaffer's F6 function $f_6(x) = g(x_1, x_2) + g(x_2, x_3) + \dots + g(x_{D-1}, x_D) + g(x_D, x_1)$ $g(x, y) = 0.5 + \frac{(\sin^2(\sqrt{x^2+y^2})-0.5)}{(1+0.001(x^2+y^2))^2}$	600
	F4	Shifted and rotated Schwefel's function $f_{10}(x) = 418.9829 \times D - \sum_{i=1}^D g(z_i)$	1000
Hybrid functions	F5	Hybrid function 5 (N = 4) $F(x) = g_1(M_1z_1) + g_2(M_2z_2) + \dots + g_N(M_Nz_N) + F^*(x), N = 4$	1500
	F6	Hybrid function 6 (N = 6) $F(x) = g_1(M_1z_1) + g_2(M_2z_2) + \dots + g_N(M_Nz_N) + F^*(x), N = 6$	2000
Composition Functions	F7	Composition function 5 (N = 5) $F(x) = \sum_{i=1}^N \{\omega_i^* [\lambda_i g_i(x) + bias_i]\} + F^*, N = 5$	2500
	F8	Composition function 8 (N = 6) $F(x) = \sum_{i=1}^N \{\omega_i^* [\lambda_i g_i(x) + bias_i]\} + F^*, N = 6$	2800

### 3.2. Auto-Tuning Performance Test

From the flowchart of the QMS auto-tuning algorithm, auto-tuning can be divided into two phases: calibration of the mass axis and resolution, and optimization of the lens and ion source parameters. Thus, in the following experiments, we used a triple QMS (Tianjin Guoke Medical Technology Development Co., Ltd., Tianjin, China), and the test sample was a polypropylene glycol PPG2000 solution with a concentration of 0.002 mol/mL and a relative molecular weight range of 20–1200. We tested these two stages, that is, the automatic calibration experiment of the mass axis and resolution, and the optimization experiment of the lens and ion source parameters, separately with correct and error-free results. The two processes were then combined and 10 tests were performed.

#### 3.2.1. Auto-Calibration Testing of Resolution and Mass Axis

To test optimization performance of the algorithm for the resolution and mass axis, we selected low, medium, and high mass numbers, and conducted resolution and mass axis tuning experiments simultaneously. Three mass numbers 59.05, 616.46, and 906.67  $m/z$  were selected to represent low-, medium-, and high-mass segments, respectively.

The QMS requires an FWHM (unit-mass resolution) range of 0.5 to 0.7  $m/z$  for low mass numbers, 0.6 to 0.8  $m/z$  for medium mass numbers, and 0.6 to 0.85  $m/z$  for high mass numbers. The mass accuracy of the full-mass segment should be no more than 0.2  $m/z$ .

The following settings were selected as the input parameters: velocity boundary  $v_{min} = (-0.5, -60), v_{max} = (0.5, 60)$ ; parameter  $\alpha = 0.1, \beta = 0.05$ ; maximum number of iterations  $N = 50$ ; weight boundary  $w_{min} = (-8, 0), w_{max} = (8, 65535)$ ; initial velocity is 0; and SA parameters  $\varepsilon = 0.5, \eta = 0.9$ , and  $T = 80$ .

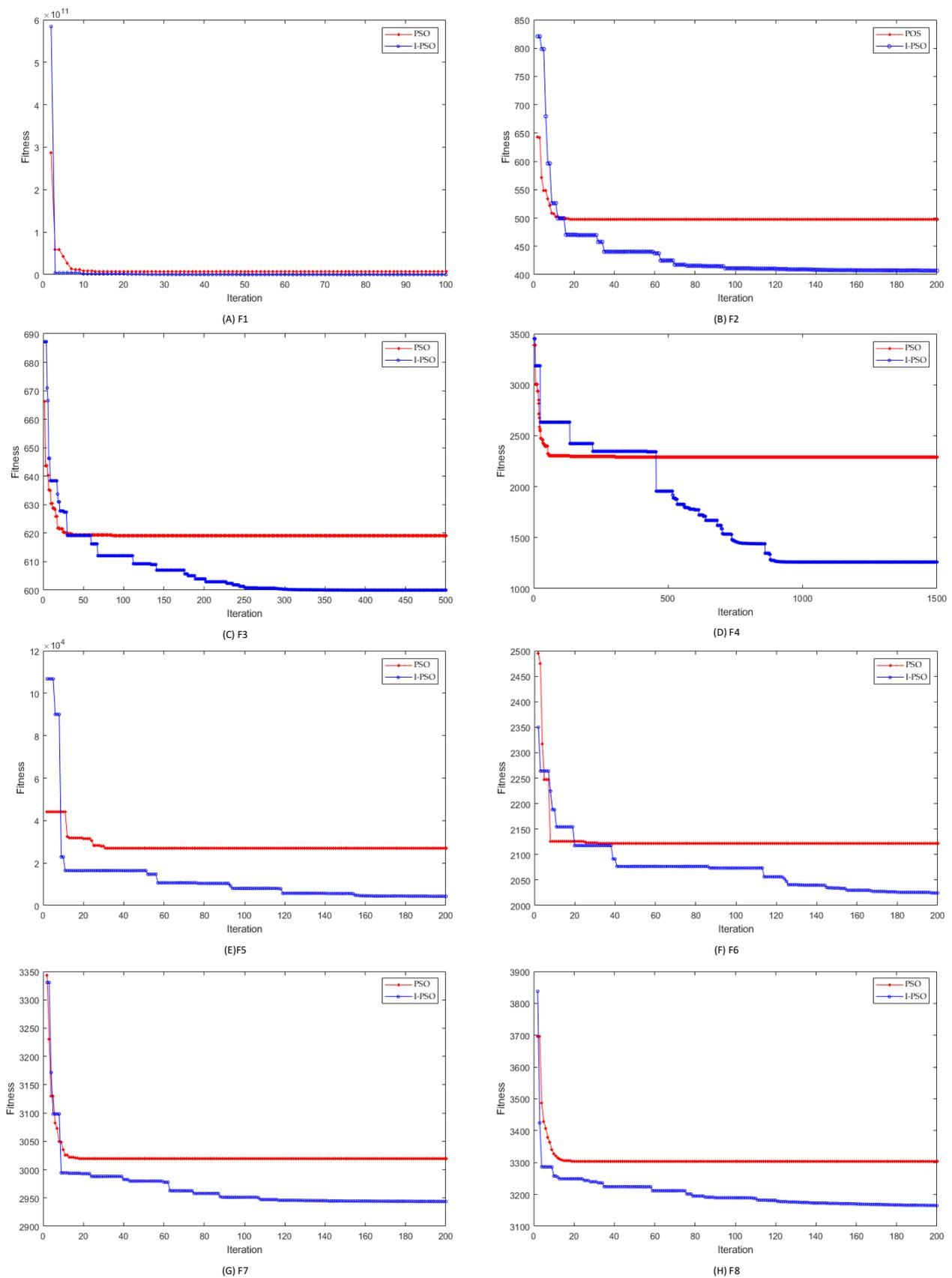
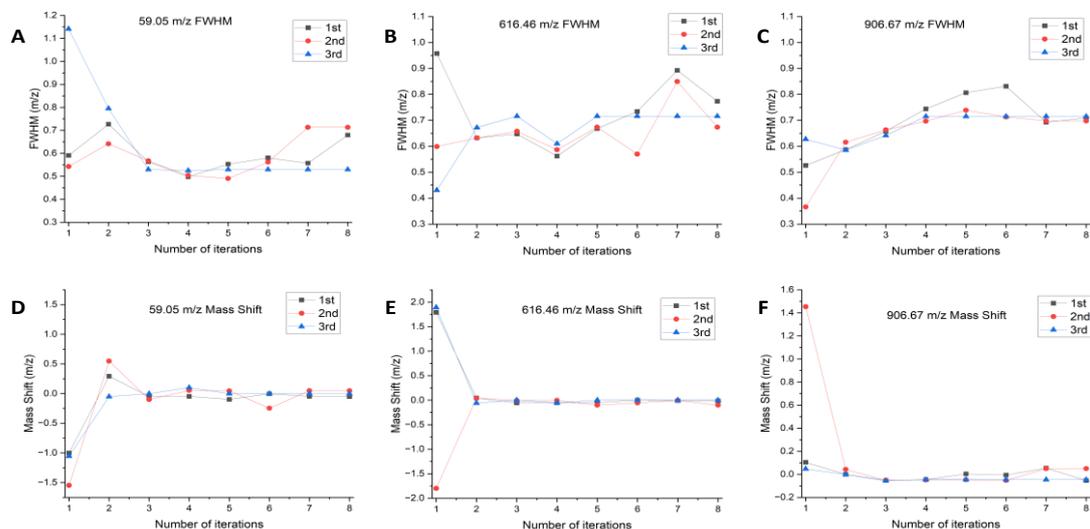


Figure 4. Comparison of the algorithm's iterative performance on eight test functions.

The mass numbers were tested with values of 59.05, 616.46, and 906.67  $m/z$ . The FWHM and mass axis of each mass number were obtained from three different initial positions, and calibration tests were performed until convergence to the required FWHM and mass accuracy range. The experimental results are shown in Figure 5.



**Figure 5.** Iterative process for resolution calibration and mass axis. (A) 59.09  $m/z$  iterative process for resolution calibration; (B) 616.46  $m/z$  iterative process for resolution calibration; (C) 906.67  $m/z$  iterative process for resolution calibration; (D) 59.09  $m/z$  iterative process for mass axis; (E) 616.46  $m/z$  iterative process for mass axis; (F) 906.67  $m/z$  iterative process for mass axis.

After analyzing the data, the mass number 59.05  $m/z$ , according to Figure 5A,D, three experiments of resolution calibration and mass axis reached the optimization at the fifth, third, and third iterations, the final output FWHMs were 0.5532, 0.5674, and 0.5297  $m/z$ , and the corresponding mass shifts were  $-0.0523$ ,  $-0.1000$ , and  $-0.0008$   $m/z$ ; the mass number 616.46  $m/z$ , according to Figure 5B,E, resolution calibration and mass axis of the three experiments were optimized at the eighth, fifth, and fifth iterations, and the final output FWHMs were 0.7399, 0.6738, and 0.7159  $m/z$ , corresponding to mass shifts of  $-0.0133$ ,  $-0.1025$ , and  $-0.0000$   $m/z$ ; and the mass number 906.67  $m/z$ , the three resolution calibration and mass axis experiments were optimized at the 8th, 4th, and 4th iterations, and the final output FWHM values were 0.7093, 0.6978, and 0.7149  $m/z$ , corresponding to mass shifts of  $-0.0541$ ,  $-0.0487$ , and  $-0.0463$   $m/z$ .

In the experimental procedure, we alternated between the resolution adjustment and mass axis calibration. Termination condition (I) was successfully satisfied, resulting in the final output values. The experiment demonstrated that both resolution and mass axis calibration could effectively meet the requirements of mass spectrometry experiments, regardless of the initial states (i.e., initial FWHM and mass axis). The difference observed in the number of iterations primarily corresponded to the spectral peaks where the FWHM and mass axis deviated significantly from the theoretical values. In such cases, a higher number of iterations is necessary to converge toward the theoretical values, which align with the fundamental principles of the algorithm.

### 3.2.2. Auto-Optimization Testing of Lens and Ion Source Parameters

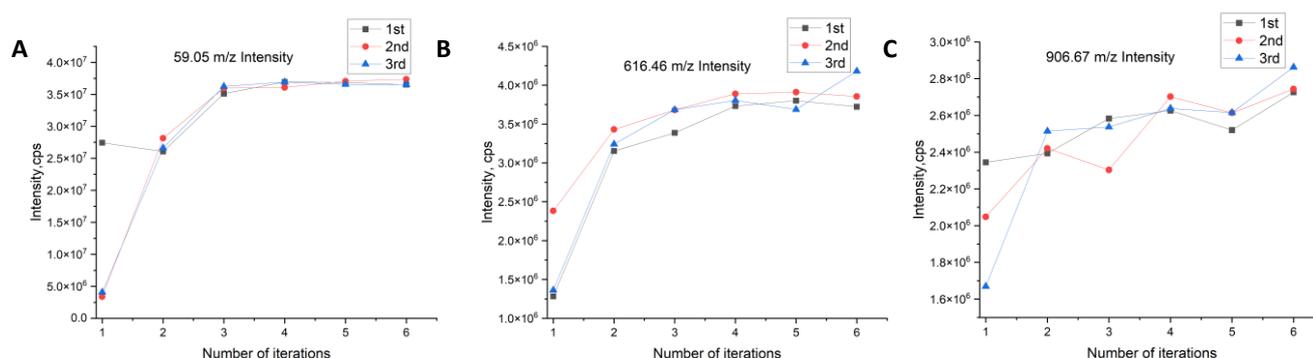
The optimization of the mass spectrometry lens and ion source parameters is based on the completion of the mass axis and resolution calibration, and we choose the output of one of the groups in Section 3.1 (mass axis parameters and resolution) as the initial conditions for the automatic optimization test of the lens and ion source parameters. In this test, we chose to optimize the parameters for low, medium, and high mass numbers and tested the mass numbers 59.05, 616.46, and 906.67  $m/z$ , with a focus on the result of interest in the test

being the spectral peak intensity. We set the passing intensity based on a priori knowledge, and during this intensity, the optimal parameters were considered to be achieved.

In this experiment, we chose the Q1POS scan mode of the mass spectrometer for testing, and four parameters were optimized; namely, curtain gas (CUR), atomization gas (GS1), declustering voltage (DP), and entrance voltage (EP).

In this paper, we form a vector of four parameters of the mass spectrometry lens and ion source parameters CUR, GS1, DP and EP, which we denote as (CUR,GS1,DP,EP), and these four parameters are tensored into a four-dimensional space. Corresponding to this four-dimensional space is Equation (12), which is in one-to-one correspondence. The following algorithm will perform iterative experiments on the vectors (CUR,GS1,DP,EP) of the four-dimensional space, expecting to obtain (CUR,GS1,DP,EP), which minimizes Equation (12).

The following settings were defined for the input parameters: velocity boundary  $v_{\min} = (-5, -5, -1, -1)$ ,  $v_{\max} = (5, 5, 1, 1)$ , where  $v_{\min}$ ,  $v_{\max}$  are the movement ranges of the vectors (CUR, GS1, DP, EP); parameters  $\alpha = 0.1$ ,  $\beta = 0.05$ , that determine the move step during the iteration of the vector (CUR,GS1,DP,EP); maximum number of iterations  $N = 50$ ; weight boundary  $w_{\min} = (0, 0, 0, 2)$ ,  $w_{\max} = (50, 90, 180, 15)$ , where  $w_{\min}$ ,  $w_{\max}$  are the boundaries of the vector (CUR,GS1,DP,EP); initial velocity is 0; and SA parameters  $\varepsilon = 0.5$ ,  $\eta = 0.9$ , and  $T = 80$ . To test the performance of the lens and ion source parameter auto-tuning algorithm, the experimental design started with different initial parameters and was then tested for convergence to the required strength. The experimental results are shown in Figure 6.



**Figure 6.** Iterative process for lens and ion source parameters. (A) 59.09  $m/z$  iterative process for lens and ion source parameters; (B) 616.46  $m/z$  iterative process for lens and ion source parameters; (C) 906.67  $m/z$  iterative process for lens and ion source parameters.

After analysis, the Q1POS mode is optimized for four parameters, and the algorithm is terminated for an average of six iterations in three experiments for each quality number. The initial parameters of the three optimization tests for each quality number were different, and the outputs of the three experiments were the same after auto-optimization, and they were all in the vicinity of the optimal solution, which achieved the expected performance.

### 3.2.3. Performance and Stability Testing

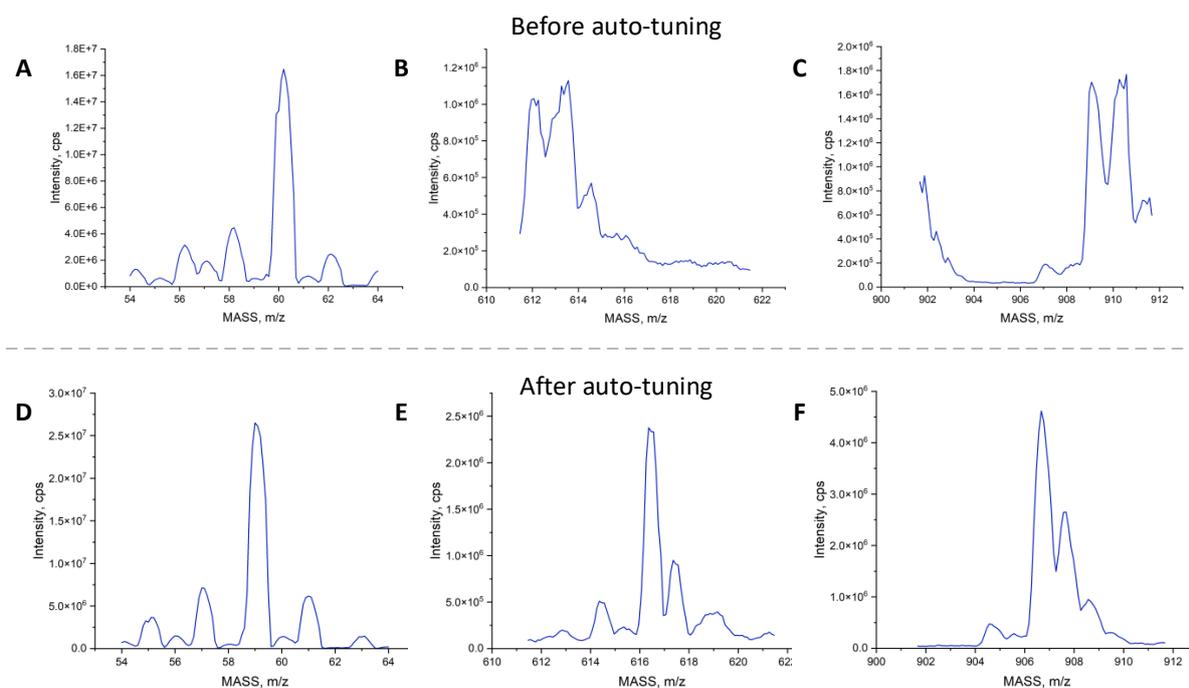
To verify the performance and stability of the algorithm, we executed the I-PSO-Tune auto-tuning algorithm several times and recorded experimental results. The experimental design started from the same initial state (mass axis and resolution parameters, lens, and ion source parameters) and was repeated 10 times; the results are shown in Table 2. The output results should satisfy the FWHM range of a low mass number: 0.5~0.7  $m/z$ , the FWHM range of a medium mass number: 0.6~0.8  $m/z$ , and the FWHM range of a high mass number: 0.6~0.85  $m/z$ . Mass accuracy in the full-mass segment should be no more than 0.2  $m/z$ .

**Table 2.** Auto-tuning repeatability experiment.

NO.	Mass Shift ( <i>m/z</i> )			FWHM ( <i>m/z</i> )			Intensity (CPS, MCA = 5)		
	59.05	616.46	906.67	59.05	616.46	906.67	59.05	616.46	906.67
1	−0.0981	0.0955	−0.0528	0.6537	0.6742	0.6569	$3.33 \times 10^7$	$2.58 \times 10^6$	$2.63 \times 10^6$
2	−0.0892	−0.1028	−0.0443	0.5950	0.6802	0.6900	$3.13 \times 10^7$	$2.98 \times 10^6$	$2.64 \times 10^6$
3	−0.0962	0.0463	−0.0952	0.6970	0.6170	0.6575	$3.19 \times 10^7$	$1.93 \times 10^6$	$2.78 \times 10^6$
4	−0.0500	0.0948	−0.1503	0.6259	0.6371	0.6170	$3.26 \times 10^7$	$2.36 \times 10^6$	$2.22 \times 10^6$
5	−0.0513	0.0942	−0.0447	0.6497	0.6298	0.6615	$3.37 \times 10^7$	$2.37 \times 10^6$	$2.71 \times 10^6$
6	−0.0489	0.0414	−0.1490	0.6239	0.7744	0.6063	$3.32 \times 10^7$	$2.74 \times 10^6$	$2.47 \times 10^6$
7	−0.0483	0.0983	−0.0495	0.6180	0.7929	0.6132	$3.06 \times 10^7$	$2.77 \times 10^6$	$2.54 \times 10^6$
8	0.0030	0.0419	−0.0510	0.5464	0.7758	0.6135	$2.96 \times 10^7$	$2.89 \times 10^6$	$2.43 \times 10^6$
9	−0.0010	0.1471	−0.0541	0.6519	0.6950	0.6408	$2.89 \times 10^7$	$2.74 \times 10^6$	$2.62 \times 10^6$
10	0.0003	0.2331	−0.0995	0.6340	0.6557	0.6177	$2.96 \times 10^7$	$2.61 \times 10^6$	$2.33 \times 10^6$

As shown in Table 2, the auto-tuning algorithm for the QMS proposed in this study can meet 100% of the mass axis and resolution calibration requirements for mass numbers 59.05 and 906.67 *m/z*, and the calibration success rate of the mass axis and resolution for mass number 616.46 *m/z* is 90%, which suggests that the parameters of our algorithm need to be further optimized. In contrast, lens and ion source parameter optimization met the requirements, with coefficient of variation (CV) values of the 10 optimal intensities for mass numbers 59.05, 616.46, and 906.67 *m/z* as 5.5, 11.9, and 6.8%, respectively. The high CV value of 616.46 was due to one set of optimized intensities being low, which may be related to the state of the mass spectrum and peak identification at that time as well as to the set a priori threshold.

Figure 7 shows the spectral signal from the mass spectrometer before and after auto-tuning. Before tuning, as shown in (A–C), the mass deviation was significant, the resolution of the spectral peaks was poor, and the peak intensity was low. However, the spectrum improved significantly after applying the auto-tuning algorithm proposed in this study, with better peak position, resolution, and intensity, as shown in (D–F). This result validates the effectiveness of the proposed algorithm in completing the instrument auto-tuning process.



**Figure 7.** Spectra before auto-tuning and after auto-tuning by I-PSO-Tune algorithm. (A–C) 59.05 *m/z*, 616.46 *m/z* and 906.67 *m/z* spectra of before auto-tuning; (D–F) 59.05 *m/z*, 616.46 *m/z* and 906.67 *m/z* spectra of after auto-tuning.

In summary, the auto-tuning algorithm for the QMS proposed in this study was demonstrated to accomplish the instrument auto-tuning process. The proposed algorithm offers several advantages.

- (1) This algorithm improves the intelligence level of the instrument, and the auto-tuning algorithm realizes the function of automatic optimization of the instrument compared with the manual tuning of the instrument, which still requires experienced engineers.
- (2) This algorithm realizes that the traditional iterative algorithm can easily fall into the local optimal solution problem from a global perspective; thus, the instrument can be automatically tuned to the real optimal state.

#### 4. Conclusions

Traditional mass spectrometer tuning relies on experienced engineers and is time consuming and inconsistent. Therefore, in this study, we propose an improved PSO auto-tuning algorithm based on the PSO algorithm by adding SA, boundary dynamics, and multiple inertia weights. The proposed algorithm solves the problem of the traditional PSO algorithm, which easily falls into a local optimal solution; thus, it is more suitable for quadrupole mass spectrometry auto-tuning. The experiments also showed that the algorithm could automatically tune an unoptimized mass spectrometer to the optimal state, and the repeatability of the algorithm was verified by repeating this experiment. The auto-tuning algorithm introduced in this study can be applied to the auto-tuning of a QMS.

In this study, the innovative auto-tuning algorithm based on the improved PSO algorithm is applied in QMS instruments to automatically adjust the parameters and make the QMS reach the optimal state. In the further work, we will try to apply this auto-tuning algorithm to other types of mass spectrometers, such as time-of-flight mass spectrometers, ion trap mass spectrometers, etc., to continuously improve the stability and versatility of the algorithm.

**Author Contributions:** Writing—original draft, M.J.; Writing—review and editing, M.J., L.L. and B.X.; formal analysis, M.J. and B.X.; funding acquisition, W.C.; methodology, L.L. and M.J.; project administration, W.C. and W.-F.D.; software, M.J., L.L. and L.F.; supervision, W.C., W.-F.D.; visualization, M.J. All authors have read and agreed to the published version of the manuscript.

**Funding:** This study was supported by the National Key R&D Program of China (grant no. 2021YFC2401100).

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

#### References

1. Douglas, D.J. Linear quadrupoles in mass spectrometry. *Mass Spectrom. Rev.* **2009**, *28*, 937–960. [[CrossRef](#)] [[PubMed](#)]
2. Cheng, Y.J.; Dong, M.; Sun, W.J.; Zhang, S.Z.; Wang, X.H.; Wu, X.M.; Zhao, L.; Chen, L.; Wu, C.Y.; Song, Y.; et al. A study of the gas interference effects in quadrupole mass spectrometer. *Meas. Sci. Technol.* **2022**, *33*, 065019.
3. Ens, W.; Standing, K.G. Hybrid quadrupole/time-of-flight mass spectrometers for analysis of biomolecules. *Method Enzymol.* **2005**, *402*, 49–78.
4. Thoren, K.L.; Colby, J.M.; Shugarts, S.B.; Wu, A.H.B.; Lynch, K.L. Comparison of information-dependent acquisition on a tandem quadrupole TOF vs. a triple quadrupole linear ion trap mass spectrometer for broad-spectrum drug screening. *Clin. Chem.* **2016**, *62*, 170–178. [[CrossRef](#)] [[PubMed](#)]
5. Perez-Arvizu, O.; Bernal, J.P. Measurement of Sulfur in Environmental Samples Using the Interference Standard Method with an O-2-pressurized Reaction Cell and a Single-Quadrupole Inductively Coupled Plasma Mass Spectrometer. *Rapid Commun. Mass Spectrom.* **2021**, *35*, e9034. [[CrossRef](#)] [[PubMed](#)]
6. Peterson, A.C.; Balloon, A.J.; Westphall, M.S.; Coon, J.J. Development of a GC/quadrupole-Orbitrap mass spectrometer, Part II: New approaches for discovery metabolomics. *Anal. Chem.* **2014**, *86*, 10044–10051. [[CrossRef](#)] [[PubMed](#)]
7. Wang, Y.W.; Prabhu, G.R.D.; Hsu, C.Y.; Urban, P.L. Tuning electrospray ionization with low-frequency sound. *J. Am. Soc. Mass Spectrom.* **2022**, *33*, 1883–1890. [[CrossRef](#)] [[PubMed](#)]
8. Kenny, D.J. Dynamic Resolution Correction of Quadrupole Mass Analyser. U.S. Patent 9,324,543, 26 April 2016.
9. Syed, S.U.; Hogan, T.J.; Antony Joseph, M.J.; Maher, S.; Taylor, S. Quadrupole mass filter: Design and performance for operation in stability zone 3. *J. Am. Soc. Mass Spectrom.* **2013**, *24*, 1493–1500. [[CrossRef](#)] [[PubMed](#)]

10. Liu, L.; Jiang, Y.; Liu, M.; Dai, X.; Fang, X.; Chen, D.; Qiu, C.; Huang, Z. Research on automatic adjustment of mass resolution in quadrupole mass spectrometry. *J. Chin. Mass Spectrom. Soc.* **2020**, *41*, 118–124.
11. Clerc, M.; Kennedy, J. The Particle Swarm—Explosion, Stability, and Convergence in a Multidimensional Complex Space. *IEEE Trans. Evolut. Comput.* **2002**, *6*, 58–73. [[CrossRef](#)]
12. Riget, J.; Vesterstrøm, J.S. *A Diversity-Guided Particle Swarm Optimizer—The ARPSO*; Department of Computer Science, University of Aarhus: Aarhus, Denmark, 2002.
13. Yu, Y.; Li, Y.; Li, J. Parameter identification of a novel strain stiffening model for magnetorheological elastomer base isolator utilizing enhanced particle swarm optimization. *J. Intell. Mater. Syst. Struct.* **2014**, *26*, 2446–2462. [[CrossRef](#)]
14. Yu, Y.; Li, Y.; Li, J. Forecasting hysteresis behaviours of magnetorheological elastomer base isolator utilizing a hybrid model based on support vector regression and improved particle swarm optimization. *Smart Mater. Struct.* **2015**, *24*, 035025. [[CrossRef](#)]
15. Bergh, F. A cooperative approach to particle swarm optimization. *IEEE Trans. Evol. Comput.* **2004**, *8*, 225–239.
16. Yue, Y.; Cao, L.; Chen, H.; Chen, Y.; Su, Z. Towards an Optimal KELM Using the PSO-BOA Optimization Strategy with Applications in Data Classification. *Biomimetics* **2023**, *8*, 306. [[CrossRef](#)] [[PubMed](#)]
17. Kennedy, J.; Mendes, R. Population structure and particle swarm performance. In Proceedings of the 2002 Congress on Evolutionary Computation, Honolulu, HI, USA, 12–17 May 2002; pp. 1671–1676.
18. Arumugam, M.S.; Chandramohan, A.; Rao, M.V.C. Competitive Approaches to PSO Algorithms via New Acceleration Co-Efficient Variant with Mutation Operators. In Proceedings of the International Conference on Computational Intelligence & Multimedia Applications IEEE, Las Vegas, NV, USA, 16–18 August 2005; pp. 225–230.
19. Kennedy, J. Why does it need velocity? In Proceedings of the 2005 IEEE Swarm Intelligence Symposium (SIS 2005), Pasadena, CA, USA, 8–10 June 2005.
20. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; IEEE Publications: Piscataway, NJ, USA, 1995; pp. 1942–1948.
21. Marini, F.; Walczak, B. Particle swarm optimization (PSO). A tutorial. *Chemom. Intell. Lab. Syst.* **2015**, *149*, 153–165. [[CrossRef](#)]
22. Du, K.-L.; Swamy, M.; Du, K.-L.; Swamy, M. *Simulated Annealing, Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature*; Birkhäuser: Cham, Switzerland, 2016; pp. 29–36.
23. Wang, Z.; Tian, J.; Feng, K. Optimal allocation of regional water resources based on simulated annealing particle swarm optimization algorithm. *Energy Rep.* **2022**, *8*, 9119–9126. [[CrossRef](#)]
24. Delahaye, D.; Chaimatanan, S.; Mongeau, M. Simulated Annealing: From Basics to Applications. In *Handbook of Metaheuristics*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 1–35.
25. Zhao, Z.; Huang, S.; Wang, W. Simplified Particle Swarm Optimization Algorithm Based on Stochastic Inertia Weight. *Appl. Res. Comput. Jisuanji Yingyong Yanjiu* **2014**, *31*, 361–391.
26. Nabi, S.; Ahmad, M.; Ibrahim, M.; Hamam, H. AdPSO: Adaptive PSO-based task scheduling approach for cloud computing. *Sensors* **2022**, *22*, 920. [[CrossRef](#)] [[PubMed](#)]
27. Xu, X.-X.; Gong, H.-L.; Ding, X.-Q. Investigation and comparison of inertia weight control schemes in particle swarm Optimization. In Proceedings of the 15th International Conference on Advanced Computational Intelligence (ICACI), Seoul, Republic of Korea, 6–9 May 2023; IEEE Publications: Piscataway, NJ, USA, 2023; Volume 2023, pp. 1–8.
28. Chroua, J.; Farhani, F.; Zaafour, A.; Jemli, M. Comparing inertia weights in multi-swarm particle swarm optimization. In Proceedings of the IEEE 2019 International Conference on Signal, Control and Communication (SCC), Hammamet, Tunisia, 16–18 December 2019; pp. 278–283.
29. Gupta, I.K.; Choubey, A.; Choubey, S. Particle swarm optimization with selective multiple inertia weights. In Proceedings of the 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, India, 3–5 July 2017; pp. 1–6.
30. Li, R.; Wang, Y. Improved particle swarm optimization based on levy flights. *J. Syst. Simul.* **2017**, *29*, 1685.
31. Kwon, Y.; Heo, S.; Kang, K.; Bae, C. Particle swarm optimization using adaptive boundary correction for human activity recognition. *KSII Trans. Internet Inf. Syst.* **2014**, *8*, 2070–2086.
32. Awad, N.H.; Ali, M.Z.; Suganthan, P.N.; Liang, J.J.; Qu, B.Y. *Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization*; Technical Report; Nanyang Technological University: Singapore, 2016.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.