mathematics

Article

Improved Particle Swarm Optimization Algorithms for Optimal Designs with Various Decision Criteria

Chang Li [1,*] and Daniel C. Coster [2]

[1] Department of Insurance, Shandong University of Finance and Economics, Jinan 250002, China
[2] Department of Mathematics and Statistics, Utah State University, Logan, UT 84322, USA; dan.coster@usu.edu
[*] Correspondence: 20207809@sdufe.edu.cn

Abstract: Particle swarm optimization (PSO) is an attractive, easily implemented method which is successfully used across a wide range of applications. In this paper, utilizing the core ideology of genetic algorithm and dynamic parameters, an improved particle swarm optimization algorithm is proposed. Then, based on the improved algorithm, combining the PSO algorithm with decision making, nested PSO algorithms with two useful decision making criteria (optimistic coefficient criterion and minimax regret criterion) are proposed . The improved PSO algorithm is implemented on two unimodal functions and two multimodal functions, and the results are much better than that of the traditional PSO algorithm. The nested algorithms are applied on the Michaelis–Menten model and two parameter logistic regression model as examples. For the Michaelis–Menten model, the particles converge to the best solution after 50 iterations. For the two parameter logistic regression model, the optimality of algorithms are verified by the equivalence theorem. More results for other models applying our algorithms are available upon request.

Keywords: improved particle swarm optimization algorithm; nested PSO algorithm; optimal experimental design; decision making criteria

MSC: 62-08

Citation: Li, C.; Coster, D.C. Improved Particle Swarm Optimization Algorithms for Optimal Designs with Various Decision Criteria. *Mathematics* **2022**, *10*, 2310. https://doi.org/10.3390/math10132310

Academic Editors: Frank Werner and Ioannis G. Tsoulos

Received: 7 May 2022
Accepted: 28 June 2022
Published: 1 July 2022

1. Introduction

1.1. Overview of PSO Algorithm

Particle swarm optimization (PSO) is a meta-heuristic, population-based swarm intelligence algorithm first proposed by [1]. Recently, the PSO algorithm has attracted a great deal of researchers for its powerful ability to solve a wide range of complicated optimization problems without requiring any assumption on the objective function. The applications of the PSO algorithm are summarized by [2,3], which divide the applications into 26 different categories, including but not limited to: image and video analysis applications, control applications, design applications, power generation and power systems, combinatorial optimization problems, etc.

The PSO algorithm is a bionic algorithm which simulates the preying behavior of a bird flock. In the Particle Swarm Optimization algorithm, each solution of the optimization problem is considered to be a "bird" in the search space, and it is called a "particle". The whole population of the solution is termed as a "swarm", and all of the particles are searched by following the current best particle in the swarm. Each particle i has two characteristics: one is position (denoted by x_i), which determines the particle's fitness value, the other is velocity (denoted by v_i), which determines the direction and distance of the search. In iteration t (t is a positive integer), to avoid confusion, the position and velocity of particle i are usually denoted by $x_i(t)$ and $v_i(t)$. Each particle tracks two "best" positions : the first is the best position found by the particle itself so far, which is denoted by "$p_{ibest}(t)$"; the second is the best position found by the whole swarm so far, denoted by "$g_{best}(t)$". When the algorithm terminates, $g_{best}(t)$ is declared to be the solution to our problem.

The velocity and position of each particle are updated by equations:

$$v_i(t+1) = \omega v_i(t) + c_1 rand_1(p_{ibest}(t) - x_i(t)) + c_2 rand_2(g_{best}(t) - x_i(t)) \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

Here, $v_i(t)$ is the velocity of the particle i, $x_i(t)$ is the position of the particle i, and ω is the inertia weight. $p_{ibest}(t)$ and $g_{best}(t)$ are the local best position for particle i and global best position found by all of the particles in iteration t, respectively. Here, $rand_1$ and $rand_2$ are two random numbers in [0, 1], while c_1, c_2 are "learning factors", with c_1 termed the "cognitive learning factor" , and c_2 the "social learning factor" [1].

From the formulas, the update of each v_i is composed of three parts: the first part is the inertia velocity before the change; the second part is the cognitive learning part, which represents the learning process of the particle from its own experience; the third part is the social learning part, which represents the learning process of the particle from the experience of other particles.

1.2. *Related Works and Main Improvements of This Manuscript*

Recent related studies for particle swarm optimization methods include: Ref. [4] proposed a multi hierarchical hybrid particle swarm optimization algorithm. Ref. [5] proposed a combination of genetic algorithm and particle swarm optimization for the global optimization. Ref. [6] proposed an improved PSO algorithm in power system network reconfiguration. Ref. [7] introduced a multiobjective PSO algorithm based on multistrategy. Ref. [8] proposed a vector angles-based many-objective PSO algorithm using archive. Ref. [9] proposed a novel hybrid gravitational search particle swarm optimization algorithm. Ref. [10] proposed the application of particle swarm optimization to portfolio construction.

Previous works provide a significant foundation for particle swarm optimization algorithms in optimal designs. However, there are some shortcomings in previous work:

(i) In order to facilitate the operation, previous works use the same dynamic parameters for the whole particle swarm in each iteration. However, in some situations, different kinds of particles may have different dynamic parameters.
(ii) Previous works mainly focus on pessimistic criterion. The combination of the PSO algorithm and other useful decision making criteria, including the optimistic coefficient criterion and minimax regret criterion, are seldom considered in previous research.

To solve these problems, in this paper, utilizing the core ideology of the genetic algorithm and dynamic parameters, an improved particle swarm optimization algorithm is proposed in Section 3. Then, based on the improved algorithm, combining the PSO algorithm with decision making, nested PSO algorithms are proposed in Section 4. The main improvements of this manuscript include: in the improved PSO algorithm in Section 3, top 10 percent particles are chosen as "superior particles". Then, different cognitive learning factors and social learning factors are used for superior particles and normal particles. In each iteration, the superior particles will split, and the inferior particles will be eliminated. In the nested PSO algorithms in Section 4, two useful decision making criteria, optimistic coefficient criterion and minimax regret criterion, are combined with the PSO algorithm. These research studies have not been conducted yet by other researchers. The details of the improvements are in Sections 3 and 4.

The proposed algorithms are implemented on various mathematical functions, which are shown in Section 5.

2. Background

2.1. *Experimental Design and the Fisher Information Matrix*

An experimental design ξ which has n support points can be written in the form:

$$\xi = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \\ \xi_1 & \xi_2 & \cdots & \xi_n \end{pmatrix}$$

Here, $x_i, i = 1 \ldots n$ are the values of the support points within the allowed design region, and ξ_i are the weights, which sum to 1, and represent the relative frequency of observations at the corresponding design point.

For several nonlinear models, the design criterion to be optimized contains unknown parameters. The general form of the regression model can be written as $y = f(\xi, \theta) + \epsilon$. Here, $f(\xi, \theta)$ can be either a linear or nonlinear function, θ is the vector of unknown parameters, and ξ is the vector of design (includes the information for both weight and the value of the support point). The range of θ is Θ, and the range of ξ is Ξ. The value of a design is measured by its Fisher information matrix, which is defined to be the negative of the expectation of the matrix of second derivatives (with respect to θ) of the log-likelihood function.

The Fisher information matrix for a given design ξ is $I(\theta, \xi) = -E[\frac{\partial^2 lnp(\xi,\theta)}{\partial\theta^2}]$. Here, $p(\xi, \theta)$ is the probability function of ξ. To estimate parameters accurately, the objective function $log|I^{-1}(\theta, \xi)|$ should be minimized over all designs ξ on Ξ.

For example, for the popular Michaelis–Menten model in biological sciences, which is presented by [11]:

$$y = \frac{ax}{b+x} + \epsilon, x > 0 \tag{3}$$

The Fisher information matrix for a given design ξ is defined by

$$I(\theta, \xi) = \int (\frac{ax}{b+x})^2 \begin{pmatrix} \frac{1}{a^2} & -\frac{1}{a(b+x)} \\ -\frac{1}{a(b+x)} & \frac{1}{(b+x)^2} \end{pmatrix} d\xi(x) \tag{4}$$

Chen et al. [12] proposed the equivalence theorem for experimental design with the Fisher information matrix: for a heteroscedastic linear model with mean function $g(x)$, $\lambda(x)$ is the assumed reciprocal variance of the response at x, then the variance of the fitted response at the point z is proportional to $v(z, \xi) = g^T(z)I(\xi)^{-1}g(z)$, where $I(\xi) = \int \lambda(x)g(x)g^T(x)\xi(dx)$. Design ξ^* is optimal if there exists a probability measure μ^* on $A(\xi^*)$, where $A(\xi^*) = \{u \in Z \mid v(u, \xi) = max_{z \in Z} v(z, \xi)\}$, Z is the range of unknown parameter. For all $x \in X$,

$$c(x, \xi^*, \mu^*) = \int \lambda(x) r(x, u, \xi^*)\mu^*(du) - v(u, \xi^*) \leq 0 \tag{5}$$

with equality at the support points of ξ^*. Here $r(x, u, \xi^*) = (g^T(x)I(\xi)^{-1}g(x))^2$

2.2. Essential Elements of Decision Making

A decision making problem is composed of four elements [13,14]:

(i) A number of actions to be taken;
(ii) A number of states which can not be controlled by the decision maker;
(iii) Objective function: payoff function or loss function which depends on both an action and a state (our objective is to maximize the payoff function or minimize the loss function);
(iv) Criterion: by certain criterion, the decision maker decides which action to take.

In the objective function $log|I^{-1}(\theta, \xi)|$, θ is in the set of states which are out of our control and design ξ is an action to be taken.

2.3. Optimization Criterion for Decision Making

Decision-making with loss functions is proposed in several papers, such as [15]. Clearly, our objective is to minimize the loss functions. Based on the loss function, there are several popular criterion for decision making:

(i) Pessimistic criterion: The pessimistic decision maker always considers the worst case, that is, suppose θ will maximize the loss function. The decision maker will take the action that minimizes the loss function in the worst case. This criterion is also known as the minimax criterion. The formula for this criterion is:

$$min_{\xi}(max_{\theta\in\Theta}log|I^{-1}(\theta,\xi)|) \tag{6}$$

(ii) Optimistic coefficient criterion: usually the decision maker will trade off from optimism and pessimism in decision making. This derives the optimistic coefficient criterion which take the weighted average of maximum and minimum of the loss function. The weight is called optimistic coefficient, which is between 0 and 1. It reflects the content of optimism of the decision maker. The formula for this criterion is:

$$min_{\xi}[(1-\alpha)max_{\theta\in\Theta}log|I^{-1}(\theta,\xi)|+\alpha min_{\theta\in\Theta}log|I^{-1}(\theta,\xi)|] \tag{7}$$

Here, α is the optimistic coefficient. When $\alpha = 0$, the optimistic coefficient criterion shrinks to pessimistic criterion.

(iii) Minimax regret criterion: in this criterion, our objective is to minimize the maximum possible regret value. The regret value is defined by the difference between the loss under certain action and the minimum loss possible under the same state. The formula for this criterion is:

$$min_{\xi}max_{\theta\in\Theta}RV(\theta,\xi) \tag{8}$$

Here, $RV(\theta,\xi) = log|I^{-1}(\theta,\xi)| - min_{\xi}log|I^{-1}(\theta,\xi)|$

The significance for criterion (ii) and (iii) are: usually the decision maker will trade off from optimism and pessimism in decision making. This derives the optimistic coefficient criterion which takes the weighted average of the maximum and minimum of the possible loss. The weight is called the optimistic coefficient, which reflects the content of optimism of the decision maker.

Some times after the decision maker makes a decision, he or she may regret when certain states appear. In this case, people want to minimize the maximum regret value, which is the distance between the loss value of the action they take and the minimum loss value possible in the relevant state. The regret value is also called opportunity costs, which represent regret in the sense of lost opportunities.

3. Improved Particle Swarm Optimization for Experimental Design

In this section, combining the core ideology of the genetic algorithm and dynamic parameters, an improved PSO algorithm is proposed as the foundation of the nested PSO algorithm.

3.1. Main Improvement of Our Algorithm

(i) The superior particles are split and the inferior particles are eliminated in each iteration. That is, in each iteration, the fitness values of the particles are ranked from high to low, and particles with top 10 percent fitness values as "superior particles" are taken. Then, each superior particle is split into two particles with the same velocities and positions, and particles with the bottom 10 percent fitness values are deleted to keep the swarm in a constant size. This improvement adopts the core ideology of the genetic algorithm: the individuals with higher fitness values will reproduce more offsprings. The splitting procedure in optimization methods is usually called individual cloning.

(ii) Dynamic parameters c_1 and c_2 are utilized in the algorithm:

$$c_1 = (c_{upper} - c_{low}) \times \frac{maxiter - iter}{maxiter} + c_{low} \tag{9}$$

$$c_2 = (c_{upper} - c_{low}) \times \frac{iter}{maxiter} + c_{low} \tag{10}$$

Here, the iter is the current number of iterations and maxiter is the maximum number of iterations. c_{upper} and c_{low} are the upper and lower bounds of the learning factors, respectively.

According to the ideology of the genetic algorithm and common sense, the superior particles have better learning abilities, thus, the learning factors for superior particles have higher upper and lower bounds than normal particles. After repeated attempts and comparisons, $c_{upper} = 2.5$, $c_{low} = 1.2$ are taken for superior particles, and $c_{upper} = 2$, $c_{low} = 0.75$ are taken for normal particles.

Consequently, in the running process of the algorithm, the cognitive learning factor is linearly decreased and the social learning factor is linearly increased.

(iii) Dynamic parameter ω is utilized in the algorithm:

$$\omega = (\omega_1 - \omega_2) \times \frac{maxiter - iter}{maxiter} + \omega_2 \tag{11}$$

ω_1 and ω_1 are the upper and lower bounds of ω, respectively. According to common sense, the superior particles are more active, thus, they have lower inertia. In our algorithm, $\omega_1 = 0.75$, $\omega_2 = 0.25$ are set for superior particles, and $\omega_1 = 0.9$, $\omega_2 = 0.4$ are set for normal particles.

Improvements (ii) and (iii) are utilized in the algorithm because these approaches are in accordance with the idea of particle swarm optimization: at the beginning, each bird has a large cognitive learning factor and small social learning factor, and each bird searches mainly by its own experience. After a period of time, as each bird gets more and more knowledge from the bird population, it relies increasingly on the social knowledge for its search. In addition, the effect of inertia velocity will decrease over time since the particles obtain more and more information from cognitive learning and social learning in the process of searching, so they rely increasingly on their learning instead of the inertia.

After applying these improvements, the improved PSO algorithms are as follows. Following the common notations in related research, in each iteration, the $x_i(t)$, $v_i(t)$, $p_{ibest}(t)$ and $g_{best}(t)$ in Formulas (1) and (2) are written as x_i, v_i, p_{ibest} and g_{best}, without causing confusion.

3.2. *Improved PSO Algorithm for Minimization/Maximization Problem*

When a maximum number of iterations is reached or when the change of the fitness value in successive searches is negligible, the stopping criteria is satisfied. In Algorithm 1, we set the maximum number of iterations to 500. When the difference of the fitness values between two adjacent iterations is less than 0.002, the change is considered as negligible.

If any value of x_i or v_i exceed the upper or lower bounds, then we will take the corresponding upper bound or lower bound instead of that value.

Clearly, this improved algorithm can be used to solve either the minimization or maximization problem. For the minimization problem, in step 1.3, the local best is the x_i with minimum $f(x_i)$. The update process of p_{ibest} and g_{best} in 2.4 is: for each particle i, if the updated fitness value is less than the fitness value of the current p_{ibest} , then p_{ibest} is updated to the new solution; otherwise, p_{ibest} remains unchanged. g_{best} is the particle which takes minimum of the fitness value of p_{ibest}.

For the maximization problem, in step 1.3, the local best is the x_i with the maximum fitness value. The update process of p_{ibest} and g_{best} in 2.4 is: for each particle i, if the updated fitness value is greater than the fitness value of the current p_{ibest} , then p_{ibest} is updated to the new solution; otherwise, p_{ibest} keeps unchanged. g_{best} is the particle which takes the maximum of the fitness value of p_{ibest} (See Figure A1 Flowchart of Algorithm 1 in Appendix A).

Algorithm 1: Improved PSO algorithm.

```
Initialization process
1.1 For each of the $n$ particles, initialize particle position $x_i$ and velocity $v_i$ with
  random values in corresponding search space.
1.2 Evaluate the fitness value of each particle $f(x_i)$ according to the
  objective function.
1.3 Determine local best and global best positions $p_{ibest}$ and $g_{best}$.
Update process:
2.1 Rank the fitness values of the particles from high to low, and take particles with
  top 10 percent fitness values as “superior particles”. Then split each superior
  particle into two particles with the same velocities and positions, and update the
  velocity of particles with Formula (1).
2.2 Based on the velocity, update the position of particles with Formula (2).
2.3 Update the fitness value $f(x_i)$.
2.4 Update the local and global best positions $p_{ibest}$ and $g_{best}$. Then update the
  fitness values of $p_{ibest}$ and $g_{best}$.
2.5 Eliminate the particles with bottom 10 percent fitness values.
If the stopping criteria is satisfied, output the $g_{best}$ and fitness value (denoted by
  $f(g_{best})$. If not, update $c_1, c_2$ and $\omega$ by Formulas (9)–(11), and repeat the
  update process.
```

4. Nested Particle Swarm Optimization for Experimental Design

The application of the Particle Swarm Optimization algorithm to the maximization and minimization problems and a nested PSO algorithm for the pessimistic criterion are presented by Chen et al. [12]. Chen’s paper is a milestone in the research of applying particle swarm optimization to experimental design. The combination of decision making and particle swarm optimization has been studied in several previous papers, such as [16,17], Yang et al. [18] and Yang and Shu [19].

However, the combination of the PSO algorithm and other decision making criteria, including the optimistic coefficient criterion and minimax regret criterion, are seldom considered in previous research. These combination problems are more interesting and challenging, and are worthy of in-depth study.

To solve these problems, nested PSO algorithms with multiple decision making criteria are proposed in this section. The implementations of these algorithms are proposed in Section 5.

4.1. Introduction of Nested PSO Algorithms

For regression with the Fisher information matrix involving unknown parameters, we need two “*swarms*” of particles (one is ξ, the other is θ) to solve it with a nested PSO algorithm. These two swarms of particles are used in different layers of iterations. In each layer, the fitness value is determined by one of the two swarms of particles. For convenience of expression, we denote the two swarms corresponding to ξ and θ by swarm 1 and swarm 2, the position by x_i and y_i, and the velocity by xv_i yv_i, respectively. Each swarm consists of 50 particles.

4.2. PSO Algorithm for Optimistic Coefficient Criterion

Define $f_{actions}(\theta,\xi) = (1-\alpha)max_{\theta\in\Theta}log|I^{-1}(\theta,\xi)| + \alpha min_{\theta\in\Theta}log|I^{-1}(\theta,\xi)|$. Our objective is to find $min_{\xi}f_{actions}(\theta,\xi)$.

When a maximum number of iterations is reached or when the change of the fitness value in successive searches is negligible, the stopping criteria is satisfied. In Algorithm 2, we set the maximum number of iterations to 100. When the difference of the fitness values between two adjacent iterations is less than 0.2 percent of the current fitness value, the change is considered as negligible.

Algorithm 2: PSO algorithm for optimistic coefficient criterion.

```
Initialization process:
1.1 For each of the $n$ particles in each of the two swarms $\xi$, and $\theta$, initialize particle
  position $x_i$, $y_i$ and velocity $xv_i$, $yv_i$ with random vectors in corresponding search
  space.
1.2 Evaluate the fitness value $max_{\theta\in\Theta}log|I^{-1}(\theta,\xi)|$ and $min_{\theta\in\Theta}log|I^{-1}(\theta,\xi)|$ by
  improved PSO algorithm. Then, initialize the $f_{actions}(\theta,\xi)$ and local and global
  best position.
1.3 Determine local best and global best positions $p_{ibest}$ and $g_{best}$.
Update process:
2.1 Rank the fitness values of the particles from high to low, and take particles with
  top 10 percent fitness values as "superior particles". Then, split each superior
  particle into two particles with the same velocities and positions, and update the
  velocity of particles by Formula (1).
2.2 Based on the velocity, update the position of particles by Formula (2).
2.3 Based on the new position, calculate the fitness value $f_{actions}(\theta,\xi)$ by
  Algorithm 1.
2.4 Update the local and global best positions $p_{ibest}$ and $g_{best}$. Then, update the
  fitness values of $p_{ibest}$ and $g_{best}$.
2.5 Eliminate the particles with bottom 10 percent fitness values.
If the stopping criteria is satisfied, output the $g_{best}$ and fitness value (denoted by
  $f(g_{best})$. If not, update $c_1, c_2$ and $\omega$ by Formula (9)–(11), and repeat the
  update process.
```

If any value of x_i, y_i or xv_i, yv_i exceed the upper or lower bounds, then we will take the corresponding upper bound or lower bound instead of that value.

In this algorithm, the process of evaluating $f_{actions}(x)$ is the inner circulation, the process of evaluating $min_\xi f_{actions}(\theta,\xi)$ is the outer circulation. Pessimistic criterion is the special case for this algorithm when $\alpha = 0$.

4.3. *PSO Algorithm for Minimax Regret Criterion*

Define $RV(\theta,\xi) = log|I^{-1}(\theta,\xi)| - min_\xi log|I^{-1}(\theta,\xi)|$. Then, this optimization problem is to find $min_\xi max_{\theta\in\Theta} RV(\theta,\xi)$. So this is a three-fold nested algorithm.

In this algorithm, the process of evaluating $f_{actions}(\theta,\xi)$ is the inner circulation, the process of evaluating $min_\xi f_{actions}(\theta,\xi)$ is the outer circulation. When a maximum number of iterations is reached or when the change of the fitness value in successive searches is negligible, the stopping criteria is satisfied. In Algorithm 3, we set the maximum number of iterations to 100. When the difference of the fitness values between two adjacent iterations is less than 0.2 percent of the current fitness value, the change is considered as negligible.

Algorithm 3: PSO algorithm for minimax regret criterion.

```
Initialization process:
1.1 For each of the $n$ particles in each of the two swarms, $\xi$ and $\theta$, initialize particle
  position $x_i$, $y_i$ and velocity $xv_i$, $yv_i$ with random vectors.
1.2 Compute the fitness value $min_{\xi} log|I^{-1}(\theta,\xi)|$ by improved algorithm. Based on
  that, compute $RV(\theta,\xi)$.
1.3 Determine local best and global best positions $p_{ibest}$ and $g_{best}$.
Update process:
2.1 Rank the fitness values of the particles from high to low, and take particles with
  top 10 percent fitness values as "superior particles". Then, split each superior
  particle into two particles with the same velocities and positions, and update
  velocity $yv_i$ of particles in swarm 2 by Formula (1).
2.2 Based on the velocity, update the position of particles in swarm 2 by
  Formula (2).
2.3 Update the fitness value $max_{\theta\in\Theta}RV(\theta,\xi))$ with Algorithm 1.
2.4 Update velocity $xv_i$ of particles in swarm 1 by Formula (1).
2.5 Based on the velocity, update the position of particles by in swarm 1
  Formula (2).
2.6 Update the fitness value (the loss function) $min_{\xi}max_{\theta\in\Theta}RV(\theta,\xi)$ by
  Algorithm 1. Then, update $p_{ibest}$ and $g_{best}$. Then, update the fitness values of
  $p_{ibest}$ and $g_{best}$.
2.7 Eliminate the particles with bottom 10 percent fitness values.
If the stopping criteria is satisfied, output the $g_{best}$ and fitness value (denoted by
  $f(g_{best})$.
If not, update $c_1$, $c_2$ and $\omega$ by Formulas (9)–(11), and repeat the update process.
```

5. Results and Comparisons

In this section, in Section 5.1 the improved PSO algorithm in Section 3 is compared with the traditional PSO on two unimodal functions and two multimodal functions to show the ability of the algorithm. In Section 5.2, we propose the comparisons of our improved PSO algorithm in Section 3 with a typical algorithm on one unimodal functions and two multimodal functions. The test functions of Sections 5.1 and 5.2 are mainly chosen from [4]. After that, the nested PSO algorithms in Section 4 are applied on two representative models with unknown parameters as examples. The algorithms are programmed by Matlab 2020a, and operated under Intel Core i7, Windows 7.

5.1. Comparisons of Traditional PSO Algorithm and Improved PSO Algorithm

In this subsection, the improved PSO algorithm in Section 3 is compared with traditional PSO on two unimodal functions and two multimodal functions. Table 1 is the information of test functions. In order to facilitate the calculation, the original minimum problems are transformed into maximum problems with maximum value = 100, that is why the fitness values are defined as $\frac{1}{F+0.01}$.

Table 1. Information of test functions.

Function	Expression	Search Space	Global Optimum	Fitness Value
Quadric	$F=\sum_{i=1}^{n}(\sum_{j=1}^{i}x_j^2)$	$(-50,50)^n$	$F=0$	$\frac{1}{F+0.01}$
Tablet	$F=10^6x_1^2+\sum_{i=2}^{n}x_i^2$	$(-50,50)^n$	$F=0$	$\frac{1}{F+0.01}$
Griewank	$F=1+\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}cos(\frac{x_i}{\sqrt{i}})$	$(-500,500)^n$	$F=0$	$\frac{1}{F+0.01}$
Rastrigin	$F=10n+\sum_{i=1}^{n}[x_i^2-10cos(2\pi x_i)]$	$(-5,5)^n$	$F=0$	$\frac{1}{F+0.01}$

To eliminate the randomness, for each function, each algorithm is run 50 times independently, and the statistical results are analyzed. In each run, when the fitness value is more than 99.9, the computation is considered as successful. In the table, the success rate indicates the percentage of success (not the times of success), and average indicates the average of the fitness values.

Table 2 is the comparisons of the performance of traditional PSO and improved PSO.

Table 2. Comparisons of the performance of traditional PSO and improved PSO.

Function	Method	Success Rate	Average
Quadric	Traditional PSO	82	85.3673
Quadric	Improved PSO	98	99.6342
Tablet	Traditional PSO	78	81.1744
Tablet	Improved PSO	100	99.8431
Griewank	Traditional PSO	74	78.0472
Griewank	Improved PSO	98	99.7956
Rastrigin	Traditional PSO	68	72.2582
Rastrigin	Improved PSO	96	99.1391

From Table 2, the results of our improved PSO algorithm are much better than that of traditional PSO for all of the four test functions, which confirmed the ability of the improved PSO algorithm.

5.2. Comparisons of Our Improved PSO Algorithm with a Typical Combination of Genetic and PSO Algorithm

Ref. [5] proposed a typical combination of the genetic and particle swarm optimization algorithm for the global optimization, which incorporates the crossover and mutation operations of the genetic algorithm into the PSO algorithm. This combination method is a common PSO variant. In this subsection, we propose the comparisons of our improved PSO algorithm with that typical combinatorial algorithm on one unimodal function and two multimodal functions to show the ability our improved algorithm. In this subsection, we use two new test functions on Tables 1 and 3 test function on Section 5.1.

Table 3. Information of new test functions.

Function	Expression	Search Space	Global Optimum	Fitness Value
Rosenbrock	$F = \sum_{i=1}^{n-1}(100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2)$	$(-2, 2)^n$	$F = 0$	$\frac{1}{F+0.01}$
Schaffer's	$F = \sum_{i=1}^{n-1}(x_i^2 + x_{i+1}^2)^{0.25}[sin(50(x_i^2 + x_{i+1}^2)^{0.1}) + 1]$	$(-100, 100)^n$	$F = 0$	$\frac{1}{F+0.01}$

The number of runs of each algorithm is the same as Section 5.1. In each run, when the fitness value is more than 99.9, the computation is considered as successful. In the table, the success rate indicates the percentage of success (not the times of success), and average indicates the average of the fitness values.

Table 4 is the comparisons of the performance of our improved PSO algorithm with the typical combination of genetic and PSO algorithm in [5].

Table 4. Comparisons of our improved PSO algorithm with a typical combination of genetic and PSO algorithm.

Function	Method	Success Rate	Average
Rosenbrock	Typical combination of genetic and PSO algorithm	90	92.1891
Rosenbrock	Improved PSO	98	99.4084
Schaffer's	Typical combination of genetic and PSO algorithm	84	89.2963
Schaffer's	Improved PSO	96	99.2861
Rastrigin	Typical combination of genetic and PSO algorithm	82	87.4295
Rastrigin	Improved PSO	96	99.1391

From Table 4, the results of our improved PSO algorithm are much better than that of the typical combination of genetic and PSO algorithm for all of the three test functions, which confirmed the ability of the improved PSO algorithm.

5.3. Implementation of Nested PSO Algorithm

In this subsection, the nested PSO algorithms in Section 4 are applied on two representative models with unknown parameters as examples. Since the most often used parameters optimistic coefficient criterion are 0.3, 0.5 and 0.7, in Tables 5 and 6, these parameters are mainly used in the computation. Extensions to other models are immediate, with a simple change of the objective function. More results of other models applying our algorithms are available upon request.

Example 1. *Michaelis–Menten model. This model and its Fisher information matrix have been introduced in Section 2. For the Michaelis–Menten model on design space $X = [0, \tilde{x}]$, Ref. [11] showed that an optimal design is supported at two support points, one of which is $\tilde{x}$. In this section, the nested algorithms in Section 4 are applied to the Michaelis–Menten model with design space $a \in [50, 100], b \in [100, 150]$, $[0, \tilde{x}]$ = [0, 200].*

Table 5. Different criterion with Michaelis–Menten model.

Criterion	$f(g_{best})$	Support Point 1	Weight 1	Support Point 2	Weight 2
Pessimistic (α = 0)	8.9996	50.1889	0.5007	200	0.4993
Optimistic coefficient with α = 0.7	5.6371	28.9594	0.5140	200	0.4860
Optimistic coefficient with α = 0.5	6.1237	91.5995	0.2158	200	0.7842
Optimistic coefficient with α = 0.3	7.5770	118.1915	0.1648	200	0.8352
Minimax regret	7.7660	39.5151	0.5648	200	0.4352

Example 2. *Two parameter logistic regression model [20]. The probability of response is assumed to be $p(x;\theta) = 1/(1 + exp(-b(x - a)))$. Here, $\theta = (a, b)^T$ is the unknown parameter vector. The information matrix of this model is:*

$$\int \begin{pmatrix} b^2 p(x,\theta)(1 - p(x,\theta)) & -b(x-a)p(x,\theta)(1 - p(x,\theta)) \\ -b(x-a)p(x,\theta)(1 - p(x,\theta)) & (x-a)^2 p(x,\theta)(1 - p(x,\theta)) \end{pmatrix} d\xi(x) \tag{12}$$

The nested algorithms in Section 4 are applied on two parameter logistic regression models with parameters $a \in [0, 2.5], b \in [1, 3], x \in [-1, 4]$.

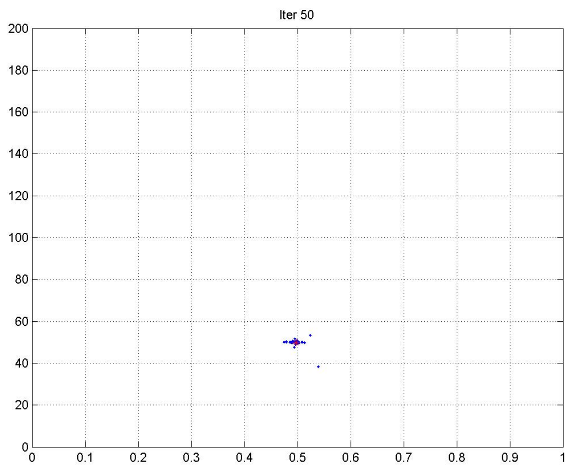
Table 6. Different criterion with two parameter logistic regression model.

Criterion	$f(g_{best})$	Support Points	Weights
Pessimistic (α = 0)	4.1104	−0.3384, 1.0064, 1.6533, 2.6503	0.2324, 0.2572, 0.2358, 0.2746
Optimistic coefficient with α = 0.7	3.3675	0.0227, 0.5731, 0.3514, −0.4051	0.2365, 0.2583, 0.2512, 0.2540
Optimistic coefficient with α = 0.5	3.4405	−0.4583, 0.6799, 0.0676, 2.2800	0.2393, 0.2459, 0.2350, 0.2798
Optimistic coefficient with α = 0.3	3.6436	2.5594, 1.6075, 2.2055, −0.2563	0.1816, 0.3006, 0.1527, 0.3651
Minimax regret	3.3282	0.6467, 1.4097, −0.2367, 0.5244	0.2376, 0.2328, 0.2308, 0.2988

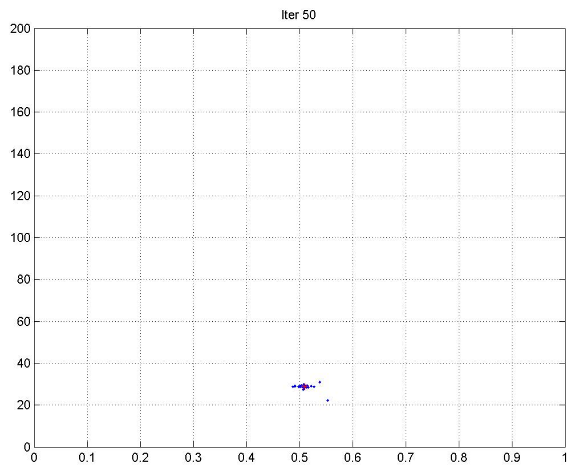
From Tables 5 and 6 above, the $f(g_{best})$ of the optimistic coefficient criterion is better than that of the pessimistic criterion , and $f(g_{best})$ is inversely proportional to α. That is because the pessimistic criterion always considers the worst case, but the optimistic coefficient criterion takes a trade off between the optimistic case and pessimistic case. When α increases, the extent of optimism gets larger, so the loss function gets smaller (and therefore better).

Figure 1 plots the value and weight of support point 1 for Michaelis–Menten model with 50 particles. Vertical coordinate represents the value of the support point 1, and horizontal coordinate represents the weight of support point 1. Figure 1 shows how the particles converges to the best solution after 50 iterations (the best solution is indicated by red star).

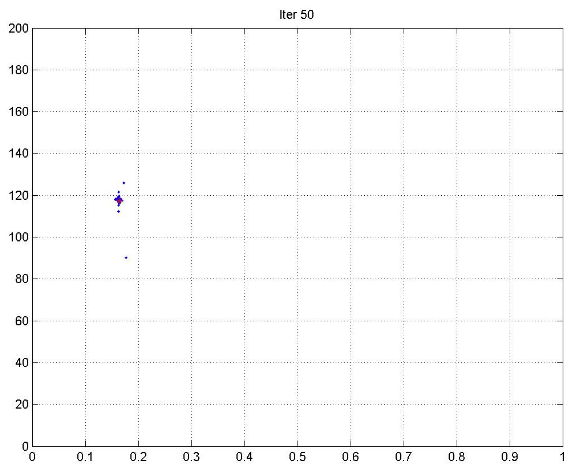
Figure 2 plots the equivalence theorem with $c(x, \xi, \mu^*)$ versus x for two parameter logistic regression models. The vertical coordinate represents the value of $c(x, \xi, \mu^*)$, and the horizontal coordinate represents the value of x. From all of these four plots, $c(x, \xi, \mu^*)$ ≤ 0 for all x $\in$ X, which confirms that all of the results obtained by our series of algorithms are optimal.



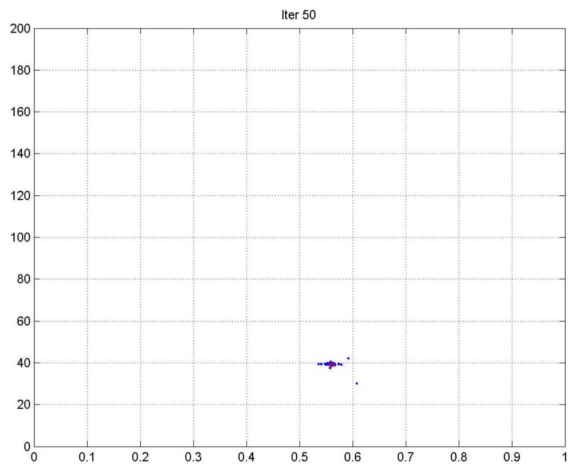
(**a**)



(**b**)

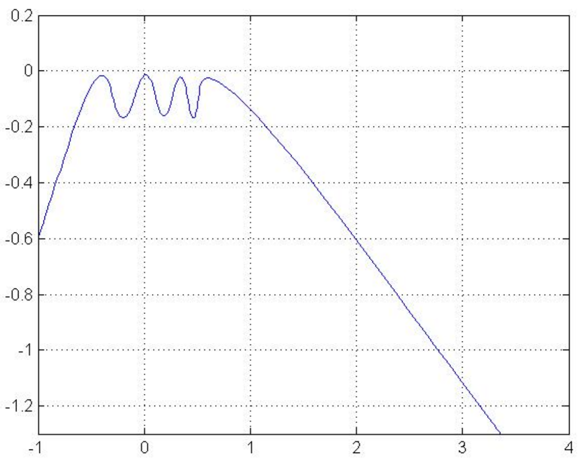


(**c**)

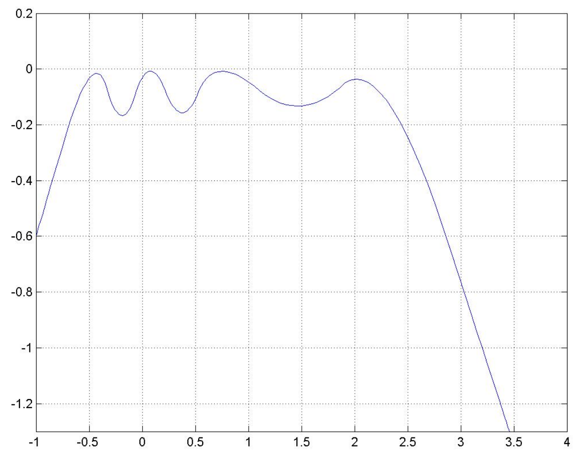


(**d**)

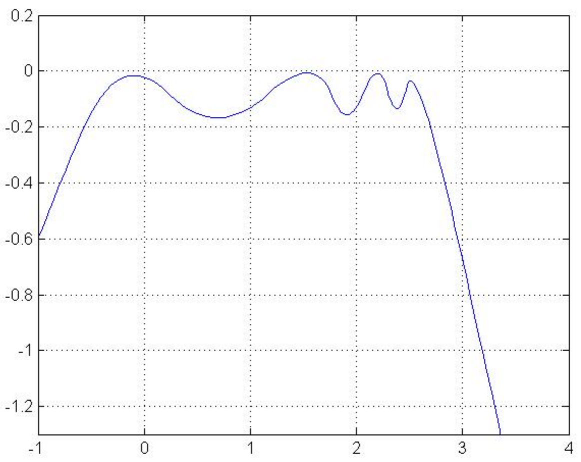
Figure 1. The convergence of particles in nested PSO algorithm for Michaelis–Menten model under various decision criteria. Vertical coordinate represents the value of the support point 1, and horizontal coordinate represents the weight of support point 1. (**a**) Michaelis–Menten model with pessimistic criterion; (**b**) Michaelis–Menten model with optimistic coefficient criterion, α = 0.7; (**c**) Michaelis–Menten model with optimistic coefficient criterion, α = 0.3; (**d**) Michaelis–Menten model with minimax regret criterion.



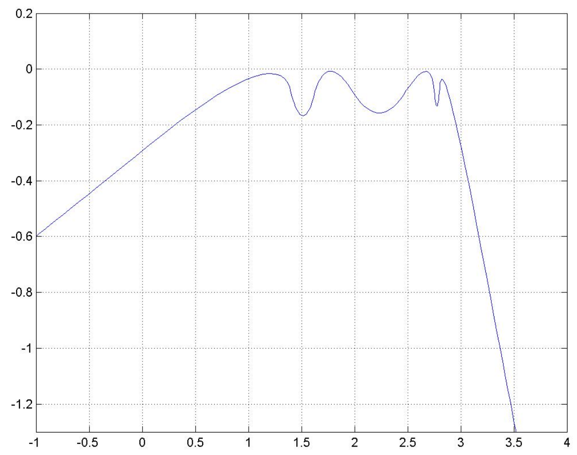
(**a**)



(**b**)



(**c**)



(**d**)

Figure 2. The plot of $c(x, \xi, \mu^*)$ versus x on nested PSO algorithm for two parameter logistic regression model. Vertical coordinate represents the value of $c(x, \xi, \mu^*)$, and horizontal coordinate represents the value of x. (**a**) Logistic regression model with optimistic coefficient criterion, α = 0.7; (**b**) Logistic regression model with optimistic coefficient criterion, α = 0.5; (**c**) Logistic regression model with optimistic coefficient criterion, α = 0.3; (**d**) Logistic regression model with minimax regret criterion.

6. Conclusions and Future Works

In this paper, an improved particle swarm optimization (PSO) algorithm is proposed and implemented on two unimodal functions and two multimodal functions. Then, combining the Particle Swarm Optimization (PSO) algorithm with the theorem of decision making under uncertainty, nested PSO algorithms with two decision making criteria are proposed and implemented on the Michaelis–Menten model and two parameter logistic regression models. For the Michaelis–Menten model, the particles converge to the best solution after 50 iterations. For two parameter logistic regression models, the optimality of algorithms are verified by the equivalence theorem. In the nested PSO algorithms, the g_{best} is inversely proportional to the optimistic coefficient.

The PSO algorithm is a powerful algorithm that needs only a well-defined objective function to minimize or maximize with different optimal criteria, here, the function $log|I^{-1}(\theta, \xi)|$. Thus, extensions to other models are immediate, with a simple change of the objective function. More results for other models applying our algorithms are available upon request. The limitation of our PSO method is: it does not work very efficiently in solving problems with a complicated matrix, which is more suitable to be solved with the simulated annealing algorithm.

Future work includes, but is not limited to, the comprehensive comparison of this series of PSO algorithms with other metaheuristic algorithms, such as the genetic algorithm and simulated annealing algorithm. This is interesting and challenging work, which will be researched in the near future.

Author Contributions: Supervision, D.C.C.; Writing—original draft, C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by NNSFC 72171133 (National Natural Science Foundation of China).

Data Availability Statement: All data supporting reported results are presented in the manuscript.

Acknowledgments: This manuscript is the improved version of the research results in Ph.D dissertation of Chang Li under the supervision of Daniel Coster in Department of Mathematics and Statistics at Utah State University. (https://digitalcommons.usu.edu/cgi/viewcontent.cgi?amp=&article=2541&context=etd) (accessed on 5 May 2022). Neither the manuscript nor any parts of its content are published in another journal. We really appreciate the editors and reviewers for their very careful reading and helpful comments.

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

Appendix A. Flowcharts of Algorithms 1–3

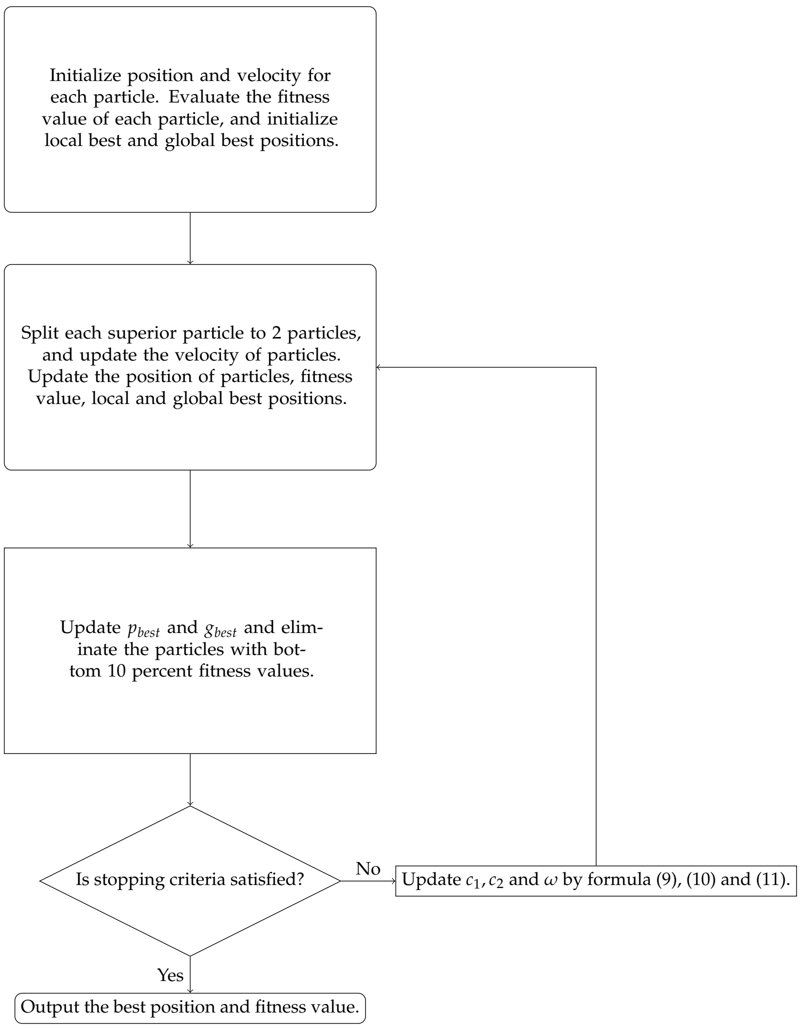

Initialize position and velocity for each particle. Evaluate the fitness value of each particle, and initialize local best and global best positions.
Split each superior particle to 2 particles, and update the velocity of particles. Update the position of particles, fitness value, local and global best positions.
Update p_{best} and g_{best} and eliminate the particles with bottom 10 percent fitness values.
Is stopping criteria satisfied?
No
Update c_1, c_2 and ω by formula (9), (10) and (11).
Yes
Output the best position and fitness value.

Figure A1. Flowchart of Algorithm 1.

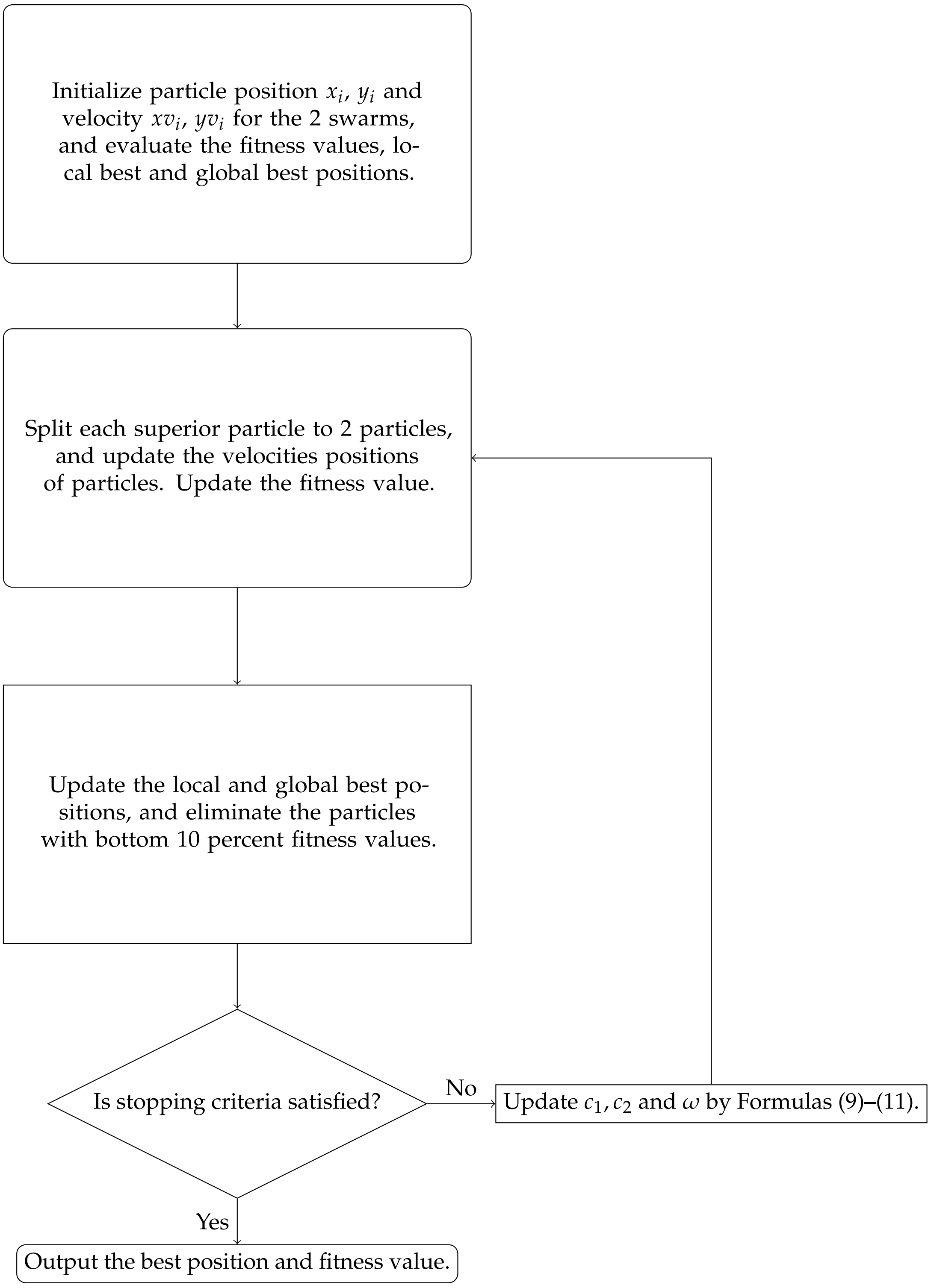

Initialize particle position x_i, y_i and velocity xv_i, yv_i for the 2 swarms, and evaluate the fitness values, local best and global best positions.
Split each superior particle to 2 particles, and update the velocities positions of particles. Update the fitness value.
Update the local and global best positions, and eliminate the particles with bottom 10 percent fitness values.
Is stopping criteria satisfied?
No
Update c_1, c_2 and ω by Formulas (9)–(11).
Yes
Output the best position and fitness value.

Figure A2. Flowchart of Algorithm 2.

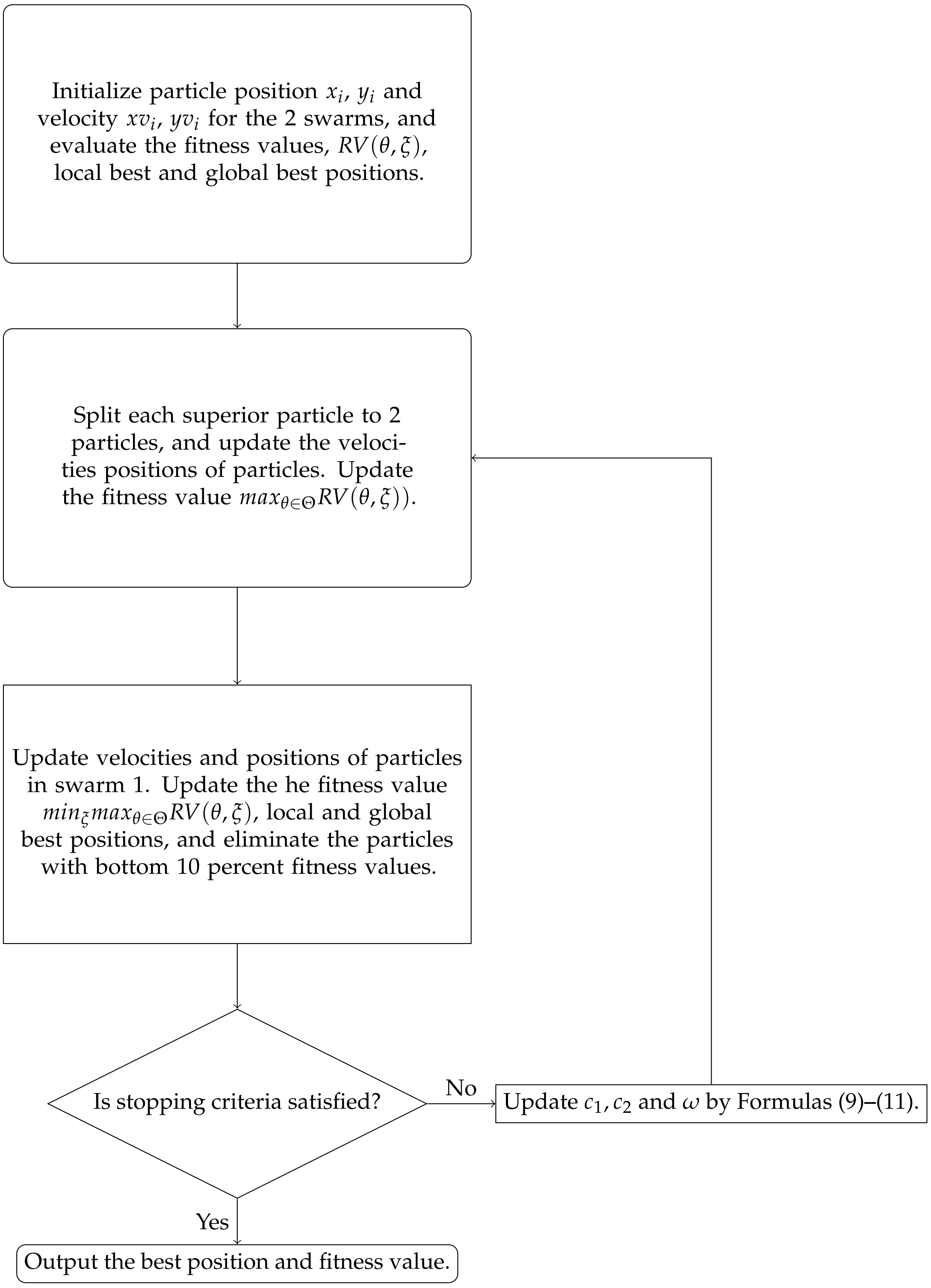

Initialize particle position x_i, y_i and velocity xv_i, yv_i for the 2 swarms, and evaluate the fitness values, $RV(\theta, \xi)$, local best and global best positions.
Split each superior particle to 2 particles, and update the velocities positions of particles. Update the fitness value $max_{\theta \in \Theta} RV(\theta, \xi))$.
Update velocities and positions of particles in swarm 1. Update the he fitness value $min_{\xi} max_{\theta \in \Theta} RV(\theta, \xi)$, local and global best positions, and eliminate the particles with bottom 10 percent fitness values.
Is stopping criteria satisfied?
No
Update c_1, c_2 and ω by Formulas (9)–(11).
Yes
Output the best position and fitness value.

Figure A3. Flowchart of Algorithm 3.

References

1. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; pp. 1942–1948.
2. Poli, R. Analysis of the publications on the applications of particle swarm optimisation. *J. Artif. Evol. Appl.* **2008**, *28*, 685175. [CrossRef]
3. Poli, R. Particle swarm optimization—An overview. *Swarm Intell.* **2007**, *1*, 33–57. [CrossRef]
4. Jin, M.; Lu, H. A multi-subgroup hierarchical hybrid of genetic algorithm and particle swarm optimization (in Chinese). *Control Theory Appl.* **2013**, *30*, 1231–1238.

5. Hachimi, H.; Ellaia, R.; Hami, A.E. A New Hybrid Genetic Algorithm and Particle Swarm Optimization. *Key Eng. Mater.* **2012**, *498*, 115–125. [CrossRef]
6. Wu, Y.; Song, Q. Improved Particle Swarm Optimization Algorithm in Power System Network Reconfiguration. *Math. Probl. Eng.* **2021**, *2021*, 5574501. [CrossRef]
7. Zou, K.; Liu, Y.; Wang, S.; Li, N.; Wu, Y. A Multiobjective Particle Swarm Optimization Algorithm Based on Grid Technique and Multistrategy. *J. Math.* **2021**, *2021*, 1626457. [CrossRef]
8. Lei, Y.A.; Xin, H.A.; Ke, L.B. A vector angles-based many-objective particle swarm optimization algorithm using archive. *Appl. Soft Comput.* **2021**, *106*, 107299.
9. Khan, T.A.; Ling, S.H. A novel hybrid gravitational search particle swarm optimization algorithm. *Eng. Appl. Artif. Intell.* **2021**, *102*, 104263. [CrossRef]
10. Chen, R.R.; Huang, W.K.; Yeh, S.K. Particle swarm optimization approach to portfolio construction. *Intell. Syst. Accounting, Financ. Manag.* **2021**, *28*, 182–194. [CrossRef]
11. Dette, H.; Wong, W. E optimal designs for the Michaelis Menten model. *Stat. Probab. Lett.* **1999**, *44*, 405–408. [CrossRef]
12. Chen, R.B.; Chang, S.P.; Wang, W.; Tung, H.C.; Weng, K.W. Minimax optimal designs via particle swarm optimization methods. *Stat. Comput.* **2015**, *25*, 975–988. [CrossRef]
13. Diao, Z.; Zhen, H.D.; Liu, J.; Liu, G. *Operations Research*; Higher Educaiton Press: Beijing, China, 2001.
14. Stoegerpollach, M.; Rubino, S.; Hebert, C.; Schattschneider, P. *Decision Making and Problem Solving Strategies*; Kogan Page: London, UK, 2007.
15. Fozunbal, M.; Kalker, T. Decision-Making with Unbounded Loss Functions. In Proceedings of the 2006 IEEE International Symposium on Information Theory, Seattle, WA, USA, 9–14 July 2006; IEEE: Piscataway, NJ, USA, 2006; pp. 2171–2175.
16. Rahmat-Samii, Y. *Evolutionary Optimization Methods for Engineering: Part II—Particle Swarm Optimization*; IEEE: Piscataway, NJ, USA, 2011.
17. Abido, A.A. Particle swarm optimization for multimachine power system stabilizer design. In Proceedings of the Power Engineering Society Summer Meeting, Vancouver, BC, Canada, 15–19 July 2001.
18. Yang, R.; Wang, L.; Wang, Z. Multi-objective particle swarm optimization for decision-making in building automation. In Proceedings of the 2011 IEEE Power and Energy Society General Meeting, Detroit, MI, USA, 24–28 July 2011; pp. 1–5.
19. Yang, L.; Shu, L. Application of Particle Swarm Optimization in the Decision-Making of Manufacturers' Production and Delivery. In *Electrical, Information Engineering and Mechatronics 2011*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 83–89.
20. King, J.; Wong, W.K. Minimax D-optimal Designs for the Logistic Model. *Biometrics* **2000**, *56*, 1263–1267. [CrossRef] [PubMed]