symmetry

MDPI

Article

An Orthogonal Multi-Swarm Cooperative PSO Algorithm with a Particle Trajectory Knowledge Base

Jun Yang, Haihua Zhu * and Yingcong Wang

College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China; nuaa.yangjun2016@gmail.com (J.Y.); dr.slark81@gmail.com (Y.W.)

* Correspondence: zhuhh@nuaa.edu.cn; Tel.: +86-158-5078-0437

Academic Editor: Vladimir Shpilrain

Received: 18 November 2016; Accepted: 13 January 2017; Published: 20 January 2017

Abstract: A novel orthogonal multi-swarm cooperative particle swarm optimization (PSO) algorithm with a particle trajectory knowledge base is presented in this paper. Different from the traditional PSO algorithms and other variants of PSO, the proposed orthogonal multi-swarm cooperative PSO algorithm not only introduces an orthogonal initialization mechanism and a particle trajectory knowledge base for multi-dimensional optimization problems, but also conceives a new adaptive cooperation mechanism to accomplish the information interaction among swarms and particles. Experiments are conducted on a set of benchmark functions, and the results show its better performance compared with traditional PSO algorithm in aspects of convergence, computational efficiency and avoiding premature convergence.

Keywords: orthogonal initialization mechanism; multi-swarm PSO algorithm; adaptive cooperation mechanism

1. Introduction

The Particle Swarm Optimization (PSO) algorithm is a stochastic, population-based evolutionary optimization technique and was conceived in 1995 by Kennedy and Eberhart [1,2]. A member of the wide category of swarm intelligence methods, the PSO algorithm is modeled on the social behaviors observed in insects, animal herds, bird flocks, and fish schools where these swarms search for food in a collaborative manner. Compared with genetic algorithm (GA), the PSO algorithm can be easily implemented and is computationally inexpensive. Each particle in the PSO algorithm represents a potential solution and aims to search a specific search space with a velocity which is dynamically adjusted according to a set of novel velocity and position updating rules reflecting a particle's own and its companions' historical behaviors. The PSO algorithm has been studied and applied to solving complex mathematical problems and real engineering optimization problems [3–6]. Nevertheless, similar to most stochastic algorithms, the performance of PSO algorithm will suffer from a significant degradation when it is applied to multi-dimensional or tightly coupled problems. To improve the performance of the traditional PSO algorithm and overcome the disadvantages of traditional PSO algorithm, different variations are presented and studied, such as hybrid PSO algorithm [7,8], multi-swarm PSO algorithm [9,10] and Quantum-behaved PSO algorithm [11].

There are some well-studied PSO algorithm performance issues such as dynamic of particle analysis [12,13], swam size [14] and swarm topology [15]. Apart from these issues, some researches focus on improving the performance of PSO by combining it with other methods [6,16]. There are also attempts to use multiple swarms to optimize different components of the solution vector cooperatively; the cooperative particle swarm optimization (CPSO) algorithm presented by Van Den Bergh and Andries [17] is a typical representative algorithm. Unlike the traditional PSO, the CPSO takes the aggregation behaviors among populations into consideration while the traditional PSO only considers

the interaction among individual particles. Moreover, Liang and Suganthan [18] also provide a similar dynamic multi-swarm particle swarm optimization algorithm, where the whole population is divided into many small swarms. New regrouping and information exchange mechanism are introduced into this model. Experiments conducted show its better performances compared with the traditional PSO algorithm and other variants.

In this paper, we present an extension PSO variant based on the CPSO model and multi-swarm PSO model mentioned above called the multi-swarm orthogonal comprehensive particle swarm optimization algorithm with a knowledge base (MCPSO-K). In this model, the searching efficiency is improved by introducing an orthogonal selection method in the population initialization phase and specific iterative node. Moreover, different from all existing PSO variants, the synchrony of flocking behavior is thought to be a function of inter-population distance and the quality of each population to maintain an optimum distance among themselves and their neighbors. The traditional interaction mechanism is replaced by a new adaptive cooperation mechanism. In addition, this MCPSO-K model is instantiated and tested by some benchmark functions, and the MCPSO-K algorithm properties are analyzed. Experiments conducted reveal advantages of the new MCPSO-K algorithm in aspects of convergence, computational efficiency and avoiding premature convergence compared with the traditional PSO algorithm.

This paper is organized as follows: Section 2 makes a brief introduction of recent researches on the PSO and other PSO variants (especially, CPSO and multi-swarm PSO, which are the ideological basis of our MCPSO-K). Section 3 describes the basic model of the MCPSO-K in detail and makes some further discussions on the MCPSO-K model. Section 4 applies the MCPSO-K model to some test functions and compares the test results with the traditional PSO algorithm. Conclusions are given in Section 5.

2. PSO and Other PSO Variants

Compared with other evolutionary algorithms, the PSO algorithm requires only primitive mathematical operators and is computationally inexpensive in terms of both speed and memory requirements. The basic problem solving processes of PSO are as follows:

Without loss of generality, an unconstrained multi-dimensional optimization problem can be formulated as a D-dimensional minimization problem expressed as follows:

$$\text{Min } f(\boldsymbol{x}), \boldsymbol{x} = [x_1, x_2, \cdots, x_D] \tag{1}$$

In order to solve this problem, the PSO algorithm first initializes a swarm consisting of m particles. The traditional PSO algorithm and most PSO variants adopt a randomized initialization mechanism, which is the reason for the unstable convergence process. Each particle in the swarm is expressed by a vector $\boldsymbol{x}_i^k = \left\{x_{i1}^k, x_{i2}^k, \cdots, x_{iD}^k\right\}, (i = 1, 2, \cdots, m)$. This vector is a point in the D-dimensional search space and means the position of the ith particle, which also represents a potential solution for the problem expressed by the Equation (1).

Then, the PSO uses the Equation (2) to update the velocity (v_{ij}^{k+1}) of each particle and Equation (3) to update the position (x_{ij}^{k+1}) of each particle in the iterative procedure. The velocity and position updating formulas are as follows:

$$v_{ij}^{k+1} = \omega \cdot v_{ij}^k + c_1 \cdot r_1 \cdot \left(p_{ij}^k - x_{ij}^k\right) + c_2 \cdot r_2 \cdot \left(p_{nj}^k - x_{ij}^k\right) \tag{2}$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1} \tag{3}$$

where the vector $v_i^k = \left\{v_{i1}^k, v_{i2}^k, \cdots, v_{iD}^k\right\}, (i = 1, 2, \cdots, m)$ is the velocity of the ith particle and k is an indicator of the population iterative process. ω is an inertia weight which not only increases the probability for algorithm to converge, but also controls the whole searching process, which is proven to

be linear or even dynamically determined by a fuzzy system [19]. c_1 and c_2 are two positive constants, and r_1 and r_2 are two random functions in the range [0, 1] and may be different for each dimension and each particle. p_{ij}^k represents the best previous position yielding the best fitness value for the ith particle; and p_{nj}^k is the best position discovered by the whole population. The relationship between these tuning parameters and the performance of the PSO algorithm are discussed in References [20–22].

By these definitions, the particles will be attracted towards the location of the best fitness achieved thus far by the particle itself and the location of the best fitness achieved thus far across the whole population. After a specific number of iterations, p_{ij}^k and p_{nj}^k will tend towards stability, where the best position can be regarded as the optimal solution of the problem. The flow chart of the traditional PSO is given in Figure 1.

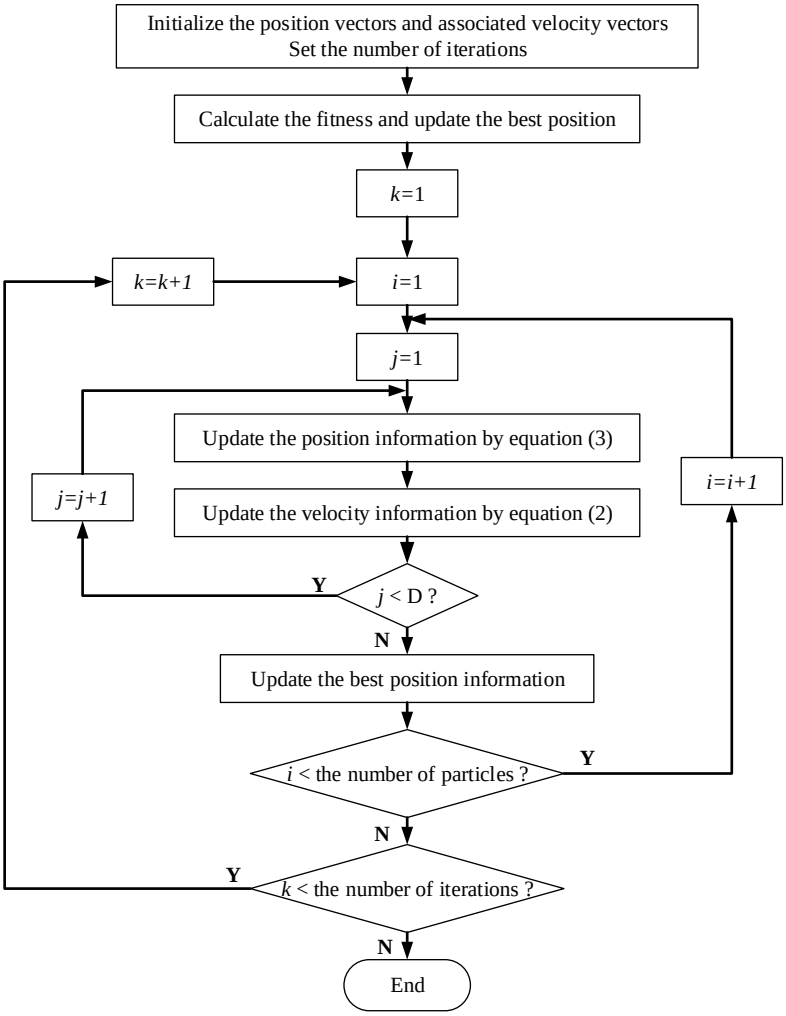

Figure 1. Flowchart of the traditional particle swarm optimization algorithm.

As mentioned above, the PSO algorithm begins as a simulation of a simplified social milieu. Based on the above characteristics, further researches were carried out to see the underlying rules that enabled large numbers of particles to search synchronously. These researches predicted that social sharing of information among particles offers the evolutionary advantage.

In order to improve the global astringency of PSO algorithm and overcome the premature convergence problem, more researchers explored various forms to realize this social information interaction. The algorithm variant based on multi-population concurrent evolution is one of the most efficiency variant. Liang et al. present a comprehensive learning particle swarm optimizer (CLPSO),

which used a novel learning strategy whereby all other particles' historical best information is used to update a particle's velocity [23]. This strategy enables the diversity of the swarm to be preserved to discourage premature convergence. Branke et al. use a number of smaller populations to track the most promising peaks over time, while a larger parent population is continuously searching for new peaks [24]. This strategy was proven to be suitable for dynamic optimization problem. Li and Yang also present a multi-swarm algorithm based on fast particle swarm optimization for dynamic optimization problems [25].

All these researches show that cooperative mode can significantly affect optimization performance and increase its ability of "exploration" and "exploitation", where exploration means the ability of a search algorithm to explore different areas of search space in order to have high probability of finding good optimum, while exploitation means the ability to concentrate the search around a promising region in order to refine a candidate solution. Most of the variants based on the PSO algorithm focus on changing or improving the "exploration" or "exploitation" ability of the algorithm and maintain the basic interaction mechanism derived from the traditional PSO algorithm. Nevertheless, most tuning parameters in traditional PSO and other variants are largely determined by prior knowledge while the determination of some parameters has a significant influence on the performance of the algorithm. Some researches show that the PSO and other variants cannot perform well for lack of prior knowledge. In other words, the PSO and most other variants do not have an adaptive solving ability for an unknown problem.

3. Multi-Swarm Orthogonal Cooperative PSO Algorithm with a Particle Trajectory Knowledge Base

3.1. Adaptive Cooperative Mechanism of Multi-Swarm Particles

Instead of changing the cooperative mode of the PSO algorithm, some variants focus on improving the performance of PSO by using different neighborhood structures [26,27]. Kennedy claimed that the PSO with large neighborhood would perform better for simple problem and PSO with small neighborhoods might perform better on complex problems [28]. Kennedy and Medas also discussed the effects of different neighborhood topological structures on the traditional PSO algorithm [29]. Hu and Eberhart proposed a dynamically adjusted neighborhood when PSO is applied into solving a multi-objective optimization problem [30]. These variants based on PSO introduce a dynamically adjusted neighborhood mechanism for each particle, where some closest particles are selected to be its new neighborhood. Different from the traditional PSO, these neighbors of a particle are all weighted and influence this particle. When these sub-groups are regarded as a small swarm, these variants can be seen as multi-swarm PSOs.

Apparently, the definition of cooperation mechanism among swarms and the regrouping method are core contents in these multi-swarm PSOs. Different from these variants above, our multi-swarm orthogonal cooperative PSO algorithm with a particle trajectory knowledge base (MCPSO-K) adopts an orthogonal initialization mechanism, which ensures the uniform distribution of swarms in initial phase and some specific iterative node in the optimization process. In addition, social sharing of information in the optimization process, which is the core content of PSO, is promoted from local level (particle level) to global level (swarm level) in MCPSO-K. Figure 2 shows a brief expression of the social sharing of information among swarms and particles in MCPSO-K.

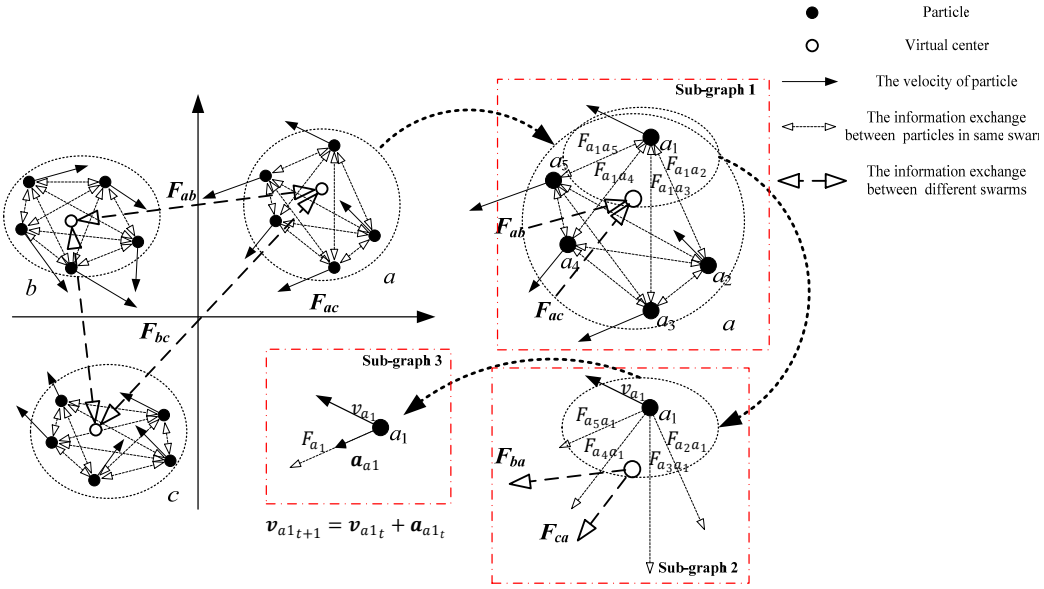

Figure 2. Social sharing of information among swarms and particles in multi-swarm orthogonal cooperative PSO algorithm with a particle trajectory knowledge base.

In Figure 2, we use three swarms (a, b, and c) with five particles in each swarm to show the social information sharing process among swarms and particles. These three swarms are distributed in the searching space uniformly using an orthogonal initialization mechanism. Each swarm has a virtual center indicating the weighted average of position and fitness value of particles in this swarm. Social information interaction in global level (swarm level) is based on this virtual center. Equations (4)–(7) give the detail quantitative method for the associated variables mentioned in Figure 2.

$$F_{ab} = \frac{k_1 \cdot d_1 \cdot f(x_{a_*}) \cdot f(x_{b_*})}{\|x_{a_*}, x_{b_*}\|^{-1}} \tag{4}$$

$$d_1 = \begin{cases} 1, \|x_{a_*}, x_{b_*}\| \geq D_{\text{swarms}} \\ -1, \|x_{a_*}, x_{b_*}\| < D_{\text{swarms}} \end{cases} \tag{5}$$

$$f(x_{a_*}) = \frac{\sum_{i=1}^{m} x_{a_i}}{m} (i = 1, 2, \cdots m) \tag{6}$$

$$f(x_{b_*}) = \frac{\sum_{i=1}^{n} x_{b_i}}{n} (i = 1, 2, \cdots n) \tag{7}$$

where F_{ab} is a quantitative indicator for the social information interaction strength between swam a and swarm b; k_1 is a constant which indicate the strength of information interaction; d_1 has only two values, which declares whether these two swarms are positive correlation or negative correlation; D_{swarms} is a predetermined parameter according to the character of problem; x_{a_i} represent position of ith particle of the swarm a; x_{a_*} and x_{b_*} are the virtual center positions of swarm a and b, respectively; the numerators $f(x_{a_*})$ and $f(x_{b_*})$ in the fraction of Equation (4) represent the fitness values of particles in swarm a and b, respectively; and the denominator of Equation (4) $\|x_{a_*}, x_{b_*}\|$ is an indicator of the distance between two swarms.

Different from most variants of PSO where particles have no mass, each particle in a swarm in MCPSO-K has mass property defined by its fitness value. Because the behaviors of swarms are determined by particles, the quantitative indicator F_{ab} defined above will ultimately act on each particle. Along with this global level (swarm level) information interaction, the local level (particle level) information interaction is defined as follows:

$$F_{a_i a_j} = \frac{k_2 \cdot d_2 \cdot f(x_{a_i}) \cdot f\left(x_{a_j}\right)}{\|x_{a_i}, x_{a_j}\|} \tag{8}$$

$$d_2 = \begin{cases} 1, \|x_{a_i}, x_{a_j}\| \geq d_{\text{swarms}} \\ -1, \|x_{a_i}, x_{b_j}\| < d_{\text{swarms}} \end{cases} \tag{9}$$

where x_{a_i} has similar implication with x_{ij}^{k+1} in Equation (3) and other parameters can be deduced from the definition in Equations (4)–(7). In order to make a clear statement of the behavior of particles, Sub-graph 1 in Figure 2 shows a partial enlarged view of the swarm a in MCPSO-K and Sub-graph 2 in Figure 2 shows a partial enlarged view of a particle in swarm a. From Sub-graphs 1 and 2, we can see that each particle has two source of influences. On the one hand, each particle is influenced by other particles belonging to a same swarm. On the other hand, all particles make up a swarm and interact with other swarms at the swarm level.

Similar to Equations (2) and (3), each particle will update its position and velocity information by updating formulas. Nevertheless, just as Sub-graph 3 shows, the velocity of a particle in MCPSO-K will be updated by the following definitions.

$$x_{ai}^{k+1} = x_{ai}^{k} + v_{ai}^{k+1} \tag{10}$$

$$v_{a_i}^{t+1} = v_{a_i}^{t} + a_{a_i}^{t} \tag{11}$$

where $a_{a_i}^t$ is defined by the following equations.

$$a_{a_i}^t = \frac{F_{a_i}^t}{f\left(x_{ai}^t\right)} \tag{12}$$

$$F_{a_i}^t = \varepsilon \cdot \sum_{j=1}^{m} F_{a_i aj}^t + \delta \cdot \sum_{\mathrm{s} \in \mathrm{S}} F_{as}^t, (\mathrm{S} = \{\mathrm{a}, \mathrm{b}, \mathrm{c}\}, i \neq j, a \neq \mathrm{s}) \tag{13}$$

Most parameters in Equations (10)–(13) are similar to the definition above. Two new parameters, ε and δ, are weight coefficients that indicate the strength of global level interactions and particle level interactions. In this way, the cooperation mechanism of MCPSO-K is accomplished by Equations (4)–(13). Different from the traditional interaction mechanism, the cooperation mechanism of MCPSO-K has the following characteristics.

(1) All particles in MCPSO-K participate in the cooperative mechanism and influence each other, while influences mentioned above are not equal strength. On the premise of not increasing the number of particles, this cooperative mechanism can largely maintain the population diversity, which includes diversity of position information and diversity of velocity information. In addition, as a result of having mass property, each particle in the MCPSO-K model can adaptively adjust its search feature according to other companions' fitness value and the distance between two particles.

(2) The MCPSO-K introduced in this paper uses the concept of a virtual center of swarm to improve the sharing of information from particle level to swarm level. In addition, particles and swarms can maintain a reasonable distance by the definition of correlation coefficient in Equations (5) and (9). A reasonable distance can be obtained by a correlation analysis in advance. In this way, MCPSO-K can largely improve its solving ability for unknown problems.

3.2. Orthogonal Initialization Mechanism

During the optimization procedure of PSO, the initial value of each particles is proven to be very important to reduce the computation time and obtain a optimum solution [19]. Determining reasonable values of each particle and optimizing the distribution of particles in the searching space have a great influence on the performance of PSO, especially in a multi-swarm cooperative PSO algorithm. While the traditional PSO algorithm and most PSO variants often adopt randomized initialization

mechanism which always causes an unstable convergence process, to solve this problem, the concept of orthogonal test design (OTD) is introduce into our MCPSO-K model in the swarm initialization phase and some specific iterative nodes.

It is well known that orthogonal test design [31] (OTD) is an efficient statistic design method. Necessary technical information about a problem can be acquired with the minimum number of tests using OTD. Design test uses OTD to determine minimum number of combinations of factor levels while these combinations are distributed over the whole space of all possible combinations. The factors are the chosen variables or parameters, which affect the chosen response variables, and each setting of a factor is regarded as a level of this factor. Nevertheless, different from the traditional OTD where specific values are set as levels of factors, value interval of a particle in each swarm in MCPSO-K model is regarded as a factor level participating in the initialization of particles in each swarm. In this way, this cooperative mechanism can largely maintain the population diversity in each generation by obtaining representative particles on the premise of not increasing the number of particles and avoiding premature convergence caused by uneven distribution of particles. Figure 3 shows the basic framework of the orthogonal initialization mechanism adopted in this paper.

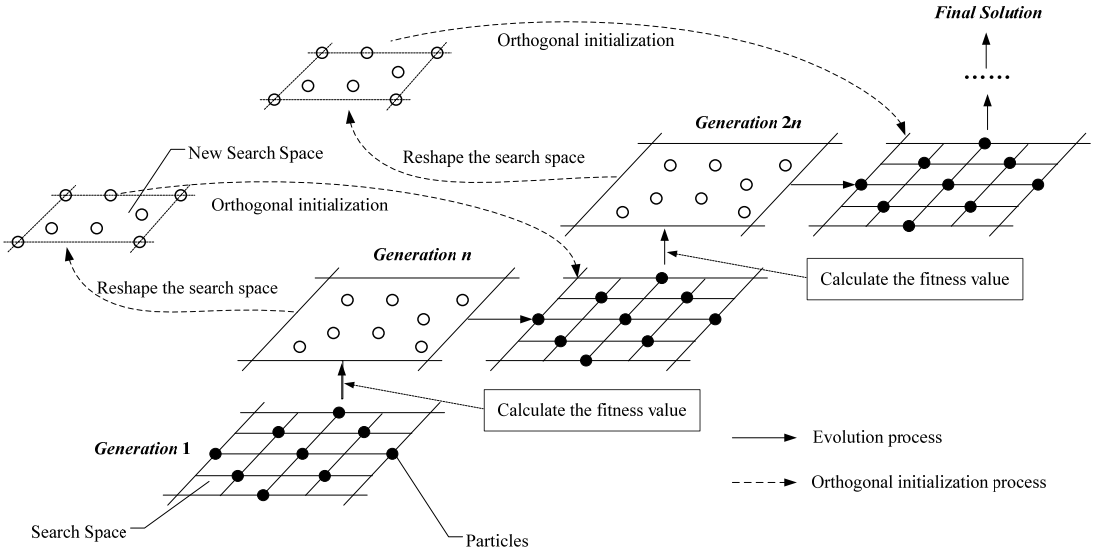

Figure 3. The framework of the orthogonal initialization mechanism.

In Figure 3, black and white roundness are used to indicate particles in different generations. We can see that OTD plays a role on the initialization phase and some specific iterative nodes (*Generation n*, *Generation 2n*, etc.) during the optimization process of MCPSO-K. Every OTD operation except the first generation is based on the value of particles obtained after some iterative operations including two sub-operations: reshaping the search space and orthogonal initialization.

(1) Sub-operation 1: Reshaping the search space

In traditional PSO algorithm, most particles will move towards the best position which always causes a particle aggregation phenomenon and ultimately influences the population diversity. In addition, to maintain a reasonable distance between particles, we also plan the local search space by this sub-operation. This operation is based on the value interval of particles ($\left[x_{ij_{min}}, x_{ij_{max}}\right]$) obtained after some iterative operations, where $x_{ij_{min}}$and $x_{ij_{max}}$ indicate the minimum and maximum value of the ith particle in j dimension, respectively. To avoid the particle aggregation phenomenon largely, an expansion factor γ is introduced into this operation. The new search space will be determined by both the expansion factor and the value interval of particles. Thus, the value interval of each dimension of the target problem can be defined as $\left[\gamma \cdot x_{ij_{min}}, \frac{1}{\gamma} x_{ij_{max}}\right], (0 < \gamma < 1)$.

(2) Sub-operation 2: Orthogonal initialization

There are many orthogonal selection techniques including Full Factorial technique, Latin Hypercube technique, Optimal Latin Hypercube technique and so on. In this paper, we adopt Optimal Latin Hypercube (OLH) technique to accomplish this orthogonal initialization. The number of levels for each factor in OLH equals the number of points with random combinations. OLH allows more points and more combinations to be studied for each factor. Engineers have total freedom in selecting the number of designs to run as long as it is greater than the number of factors.

In order to decrease the computational cost and plan sub search spaces reasonably, the initial search space of each swarm in MCPSO-K is different. Figures 4–6 show three different partition methods for ith particle domain.

Figure 3 shows the first partition method for particle domain where the ith component of the position vector $\boldsymbol{x}_i = \{x_{i1}, x_{i2}, \cdots, x_{im}\}$ of particles in swarm $j(j = 1, 2, \cdots, n-1)$ should be valued in the interval $\left[f_{ij_{min}}, f_{ij_{max}}\right]$. Some other partition methods are shown in Figures 4 and 5.

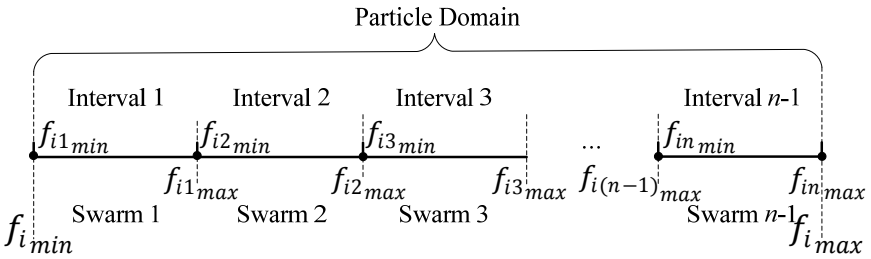

Figure 4. The first partition method for particle domain.

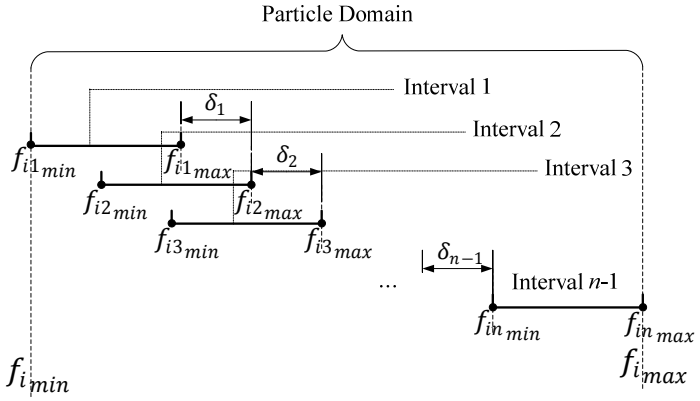

Figure 5. Second partition method for particle domain.

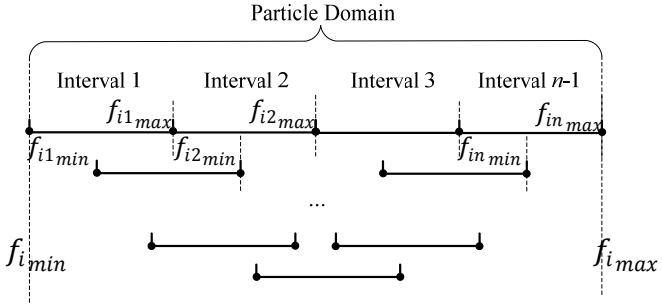

Figure 6. Third partition method for particle domain.

It is worth noting that the impact of size of overlap region δ_i in Figure 4 and the relation of $f_{ij_{max}}$ and $f_{i(j+1)_{min}}$ on performance of MCPSO-K are also a good research topic for future research.

The determination of these tuning parameters can be determined by performing sensitivity analysis on the target problem.

3.3. Particle Trajectory Knowledge Base

The traditional PSO algorithms and other variants all focus on the best positions achieved so far by the particle itself and the position of the best fitness achieved so far across the whole population. During the iterative procedure, trajectories of particles can cross each other or come into close proximity in the searching space, which means that the fitness function will be computed repeatedly. This is unacceptable in a complex problem, especially in solving some real engineering problems where the fitness value changes continuously and near positions cannot cause a significant difference in fitness value. Thus, the concept of recording the trajectory information is a good way to avoid repeated computation and is introduced into the MCPSO-K model.

As shown in Figure 7, where we use black roundness to indicate particles. The flow chart of the accomplishment of the particle trajectory knowledge base can be divided into three phases. First, an empty particle trajectory knowledge base indicated by an empty matrix is created by the information of the initialization of particles. Second, the knowledge base will record the information of the particles' position and fitness value. Third, before next iterative computation, particles that have similar position information with records in the knowledge base will be assigned the fitness value in the database directly. Note that the second and third phases are carried out simultaneously and this particle trajectory knowledge base is always expressed by a multiple dimensional matrix, where the dimension of this matrix depends on the character of a specific problem.

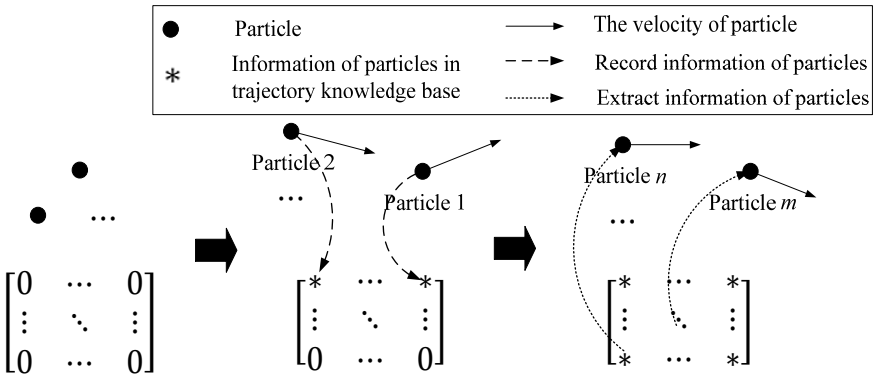

Figure 7. Particle trajectory knowledge base.

Whether to execute the assignment operation depends on the similarity between particles and records. Thus, the similarity evaluation between particles and the records in the database is a key step for this method. The similarity evaluation functions, based on the error between the position of current particle and the records in the database, are defined as follows.

$$F_{\text{similarity}} = \begin{cases} 1, f_{\text{similarity}} \leq e_{\text{similarity}} \\ 0, f_{\text{similarity}} > e_{\text{similarity}} \end{cases} \tag{14}$$

$$f_{\text{similarity}} = \sum_{i=1}^{D} \omega_i \cdot \sqrt{\left(x_{ij}^k - x_i'\right)^2} \tag{15}$$

where $F_{\text{similarity}}$ in Equation (14) has only two values indicating the similarity hypothesis holds or not, based on the current similarity level $f_{\text{similarity}}$ and a predetermined similarity level $e_{\text{similarity}}$; x_{ij}^k, which is similar to the definition in Equation (2), and x_i' presents corresponding position of previous particles recorded in the trajectory database. ω_i indicates the sensitivity of the corresponding position and the fitness value.

On the concrete operation process, this trajectory knowledge base can be expressed as a multi-dimensional matrix. It is worth noting that our study shows that implementing a sensitivity analysis based on finite number of computations before creating this particle trajectory knowledge base will contribute to the determination of the matrix dimension. In general terms, we expand spacing between adjacent values for those insensitive parameters and cut down spacing between adjacent values for those sensitive parameters.

The flow chart of the whole optimization process based on our MCPSO-K model is summarized in Figure 8.

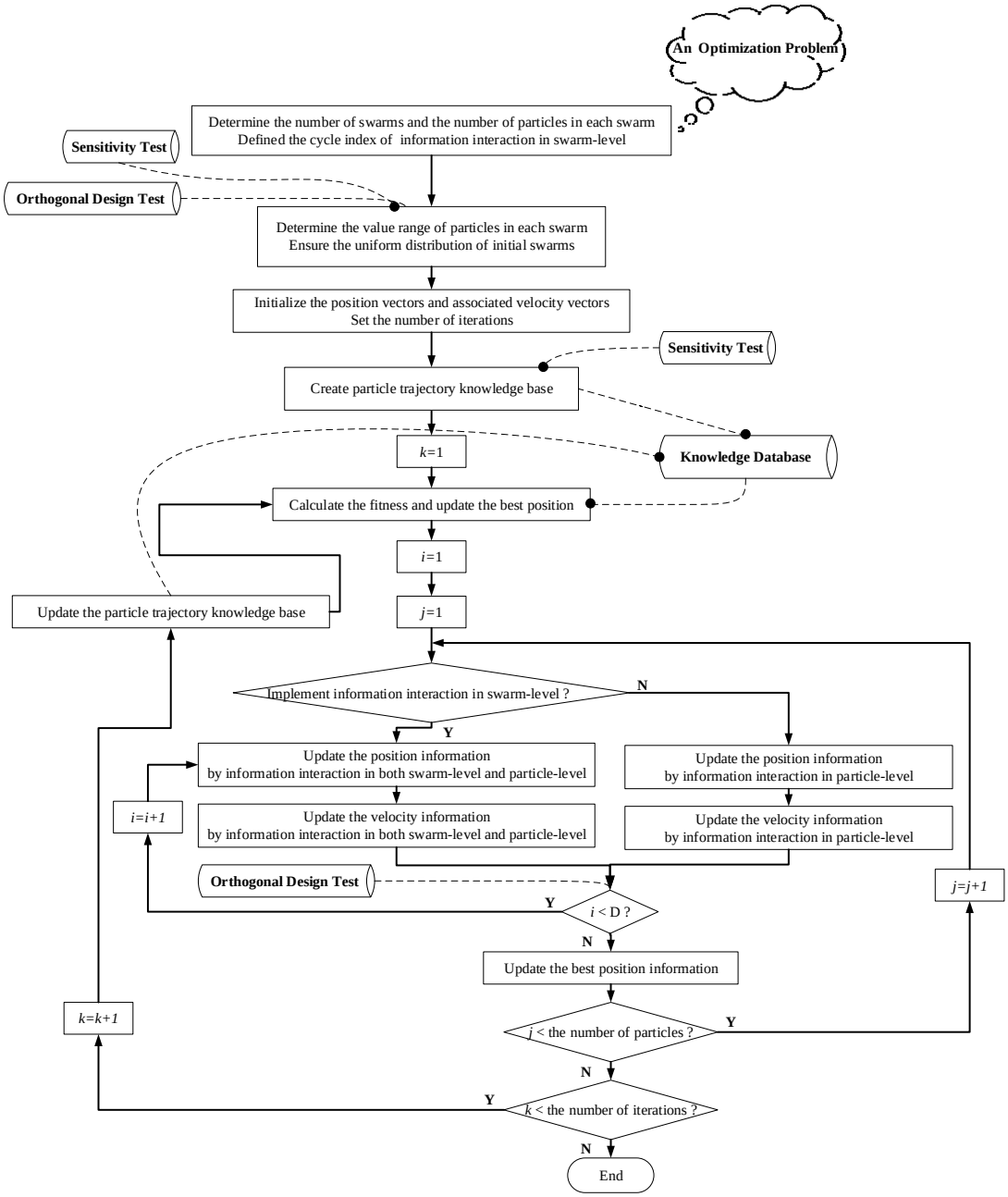

Figure 8. Flowchart of MCPSO-K algorithm.

3.4. Further Discussion about the MCPSO-K Model

Firstly, in order to dynamically adjust the ability of exploration and exploitation during searching procedure, it is beneficial to maintain a frequent information interaction among those swarms dispersed

in the search space in the beginning of the optimization. As the optimization process progresses, it is more beneficial to decrease the frequency of information interaction in swarm level and increase the frequency of information interaction in particle level, so that the swarms could effectively obtain an optimum in a local space. In fact, the CPSO-S_K and CPSO-H_K presented by Van Den Bergh and Andries also adopt some form of shared memory to build the context vector, and it is hypothesized that this vector does not have to be updated during every cycle for the algorithm to work well. Thus, the information interaction mechanism in the swarm level in our MCPSO-K should be dynamically adjusted while the information interaction mechanism in the particle level participates during the whole optimization process.

Secondly, no matter what method is used, the increase of the number of swarms and particles inevitably aggravates the computational burden. To determine a reasonable quantity of particles in each swarm is another way to reduce computation cost without influencing the performance of MCPSO-K algorithm.

Thirdly, each particle in PSO represents a complete vector that can be used as a potential solution. Each update operation in optimum iterative procedure is also performed on a full dimensional vector. This allows for the possibility that some components in the vector have moved closer to the best solution, while others actually moved away from the best solution. In this MCPSO-K model, we discuss a new updating mechanism where each swarm in MCPSO-K only makes a frequent update on some components in the vector and update other components through the information interaction in the swarm-level.

These discussions above will be considered in the implementation process of MCPSO-K in Table 1 and will be discussed briefly based on the experiments and analysis in the subsequent contents. To reduce the number of uncertainty factors in the MCPSO-K, the information exchange frequency in swarm level is used as stopping criteria for many operations in the optimization process.

Table 1. The pseudocode of the implementation process of MCPSO-K.

The Implementation Process of MCPSO
n: the number of swarms
p: each swarm's population size
h: information exchange frequency in swarm level and specific iterative node for orthogonal initialization
Max_gen: max number of generations for stopping criteria
$e_{\text{similarity}}$: evaluation index for similarity between particles
$\varepsilon_{similarity}$: adaptive index for similarity evaluation index
$g = [g_1, g_2, \cdots, g_m]$: the number of partition in each dimension
Implementation sensitivity analysis on a m dimensional optimization problem
Determine the value of $g = [g_1, g_2, \cdots, g_m]$
Determine an orthogonal combination of parameters value interval based on the sensitivity analysis
Create n swarms and each swarm has p particles
Initialize $n \times p$ particles and create a trajectory database base on the fitness value of particles
For $j = 1$ to h
For $i = 1$ to $j \times Max\_gen/h$
For $i = 1$ to n
Update the position of each particle in swarm i and calculate the fitness value
Calculate the similarity index for each particle
If the similarity index calculated above $\leq \left(h \times \varepsilon_{similarity} \times e_{\text{similarity}}\right)/j$
Update the fitness value of particle by the trajectory database
End
Update the trajectory database
End
Update the position of each particle by the cooperative mechanism defined by Equations (4)–(13)
Execute the orthogonal initialization operation
Update the best fitness of n swarms
End
End
Output the best fitness value

It must be said that, to streamline the process, h indicates the information exchange frequency in swarm level and can be regarded as a specific iterative node for orthogonal initialization.

4. Experiments and Analysis

4.1. Test Functions

We employ different types of benchmark functions, expressed as Equations (16)–(20), to test and compare the performance of the MCPSO-K algorithm we proposed. Table 1 lists the parameters used for these experiments, where the "iteration" column and "threshold" lists two normal stopping criteria for iteration and their base value.

Rosenbrock function (unimodal)

$$f_1(\boldsymbol{x}) = \sum_{i=1}^{\frac{n}{2}} \left(100 \left(x_{2i} - x_{2i-1}^2 \right)^2 + (1 - x_{2i-1})^2 \right) \tag{16}$$

Quadric function (unimodal)

$$f_2(\boldsymbol{x}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2 \tag{17}$$

Ackley function (multimodal)

$$f_3(\boldsymbol{x}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \left(\sum_{i=1}^{n} x_i^2 \right)} \right) - \exp\left(\frac{1}{n} \left(\sum_{i=1}^{n} \cos(2\pi x_i) \right) \right) + 20 + e \tag{18}$$

The generalized Rastrigin function (multimodal)

$$f_4(\boldsymbol{x}) = \sum_{i=1}^{n} \left(x_i^2 - 10 \cos(2\pi x_i) + 10 \right) \tag{19}$$

The generalized Griewank function (multimodal)

$$f_5(\boldsymbol{x}) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1 \tag{20}$$

In addition, we assume that ranges of all independent variables are determined at the interval $[-100,\ 100]$. All other relevant parameters of these functions are shown in Table 2.

Table 2. Parameters used for experiments.

Function	n	Stopping Criteria	
		Iteration	Threshold
$f_1(\boldsymbol{x})$	20	500	100
$f_2(\boldsymbol{x})$	20	500	0.01
$f_3(\boldsymbol{x})$	20	500	5.00
$f_4(\boldsymbol{x})$	20	500	100
$f_5(\boldsymbol{x})$	20	500	0.1

4.2. Test on The Number of Computation on Similar Positions

The purpose of introducing a particle trajectory knowledge base is to reduce the computation cost on similar positions of different particles during the whole iterative procedure. The number of computations on similar positions of these functions defined above is tested to prove the necessity of

the trajectory knowledge base. Table 3 lists the average number of similar computations and identical computations during iterative procedure for each test function during 10 runs. We evaluate similar and identical computations of the function defined as Equations (14) and (15) and collect data on five kinds of similarity levels using traditional PSO algorithm. As a result of different convergence generations, the number of inefficient computations are collected until the convergence. The denominator in each column of Table 3 is the number of efficient searching calculations.

Table 3. The number of inefficient computations.

Function	Similar Computation			Identical Computation	
Criteria	**≤10**	**≤5**	**≤1**	**≤0.5**	**≤0.1**
$f_1(x)$	342.1/5010	232.7/5010	55.8/5010	33/5010	0/5010
$f_2(x)$	1055.5/10,020	683.6/10,020	19.7/10,020	0.8/10,020	0/10,020
$f_3(x)$	362.6/20,020	560.2/20,020	17.8/20,020	440.2/20,020	0.4/20,020
$f_4(x)$	1424/6000	1719.8/6000	490.3/6000	756.3/6000	0.7/6000
$f_5(x)$	1775.9/10,020	1198.8/10,020	402.4/10,020	352/10,020	0/10,020

In Table 3, the evaluation "*criteria*" can also be described as a variable range Δx of each parameter, where their relations meet the following equation:

$$\sqrt{n \times \Delta \boldsymbol{x}^2} = criteria \tag{21}$$

where "*criteria*$\leq$ 5" also means the variable range $\Delta \boldsymbol{x} \leq \sqrt{25/20}$. Such a small value fluctuation of parameters will not have a significant impact on the fitness value; that is, it will not have a significant effect on the fitness value of particles.

4.3. Reliability and Accuracy of Algorithms

The reliability of algorithm is an important evaluation index for performance. The sensitivity to the quality of initialization is a significant aspect of this reliability index. The fitness value ranges of each test function in 10 representative runs after specific iterations are presented in Figures 9–18. Figures 9–13 show the fitness value range using traditional PSO algorithm and Figures 14–18 show the fitness value range using MCPSO-K algorithm. Table 4 lists the fluctuation of the historical convergence data of test functions in 10 runs using PSO. Table 5 lists the best final convergence value of each test function in 10 representative runs in 50 runs using PSO. Table 6 lists the fitness value of test functions using MCPSO-K at the same iteration. Table 7 lists the best final convergence value with MCPSO-K. It should be noted that, to make a reasonable comparison of traditional PSO and MCPSO-K, the number of swarms and particles in each swarm are equal. Thus, the number of iterations can be regarded as a good evaluation index for the performance of the algorithms.

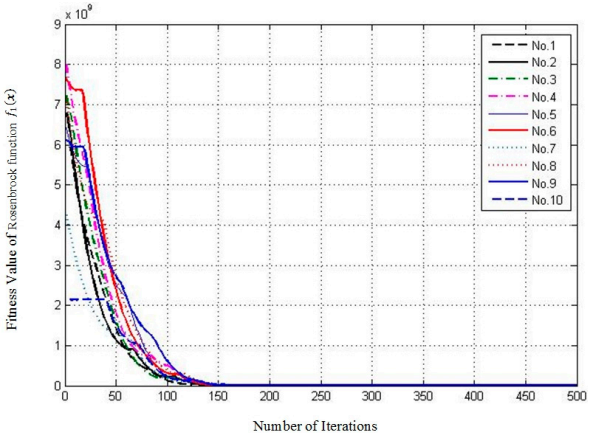

Figure 9. The fitness value of the Rosenbrock function during 10 runs using PSO.

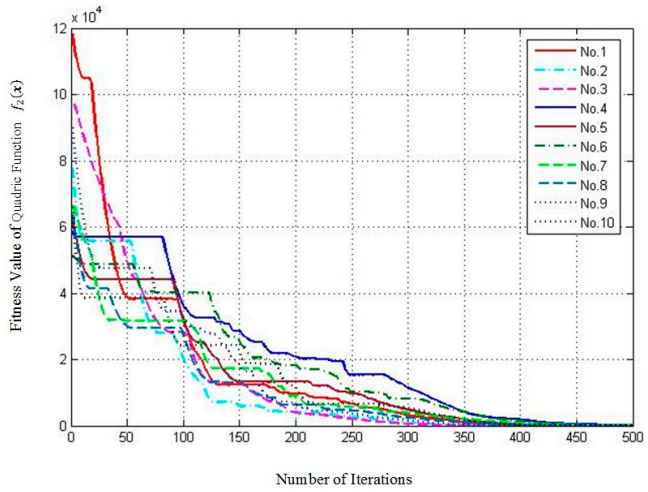

Figure 10. The fitness value of the Quadric function during 10 runs using PSO.

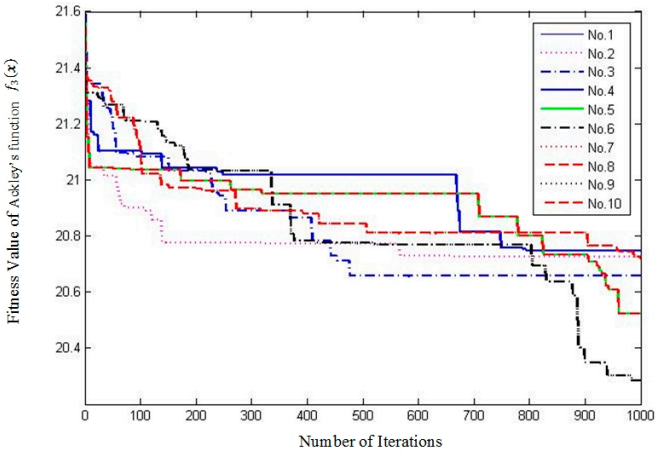

Figure 11. The fitness value of the Ackley's Function during 10 runs using PSO.

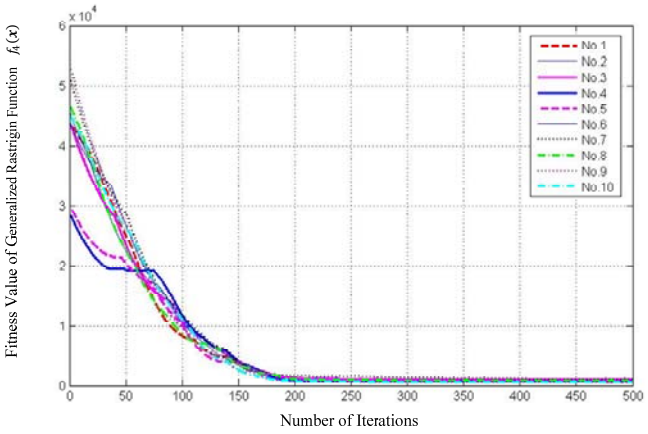

Figure 12. The fitness value of the generalized Rastrigin function during 10 runs using PSO.

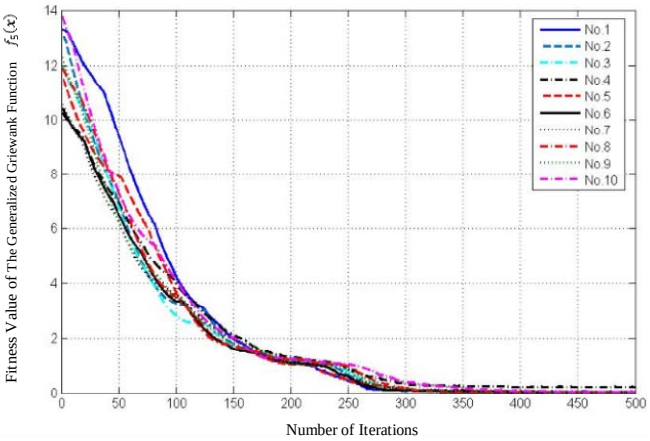

Figure 13. The fitness value of the generalized Griewank function during 10 runs using PSO.

Table 4. The fluctuation of historical convergence data during 10 runs using PSO.

Function	ITERATION		
	50	**150**	**250**
$f_1(\boldsymbol{x})$	$1.13 \times 10^9 \sim 2.71 \times 10^9$	$3.07 \times 10^6 \sim 3.43 \times 10^7$	$1.95 \times 10^4 \sim 2.21 \times 10^5$
$f_2(\boldsymbol{x})$	$3.00 \times 10^7 \sim 5.70 \times 10^7$	$6063 \sim 2.86 \times 10^7$	$2093 \sim 15360$
$f_3(\boldsymbol{x})$	$21.01 \sim 21.28$	$20.78 \sim 21.13$	$20.78 \sim 21.03$
$f_4(\boldsymbol{x})$	$1.92 \times 10^4 \sim 2.88 \times 10^4$	$2629 \sim 4114$	$614.8 \sim 1598$
$f_5(\boldsymbol{x})$	$6.22 \sim 9.38$	$1.61 \sim 2.11$	$0.45 \sim 1.03$

Table 5. The best final convergence value using PSO.

Function	The Final Convergence Value
$f_1(\boldsymbol{x})$	6005
$f_2(\boldsymbol{x})$	5.138
$f_3(\boldsymbol{x})$	20.28
$f_4(\boldsymbol{x})$	551.7
$f_5(\boldsymbol{x})$	0.0103

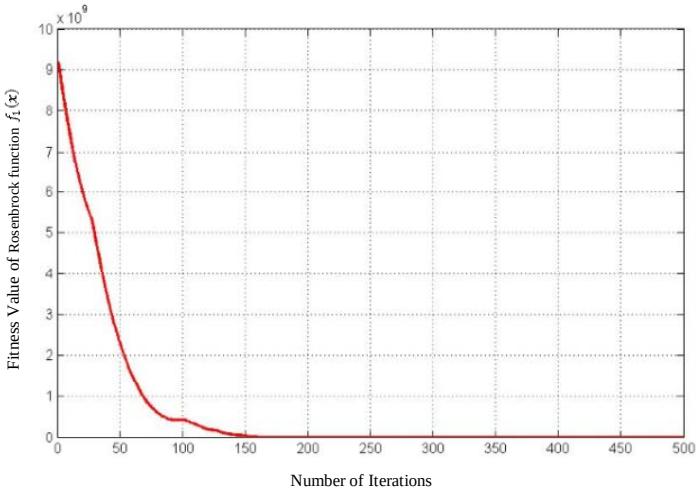

Figure 14. The fitness value of the Rosenbrock function using MCPSO-K.

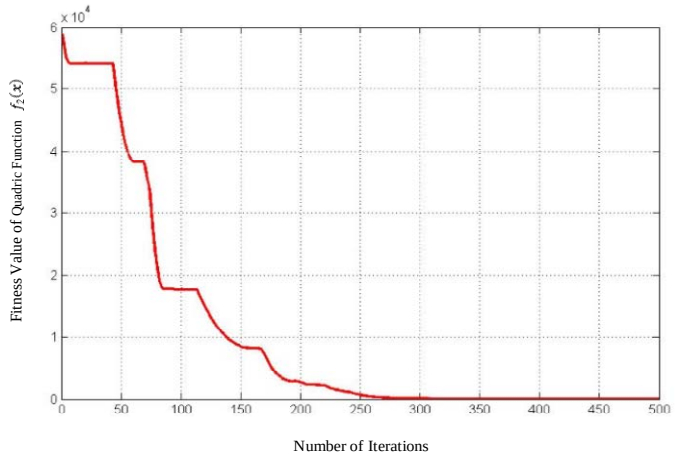

Figure 15. The fitness value of the Quadric function using MCPSO-K.

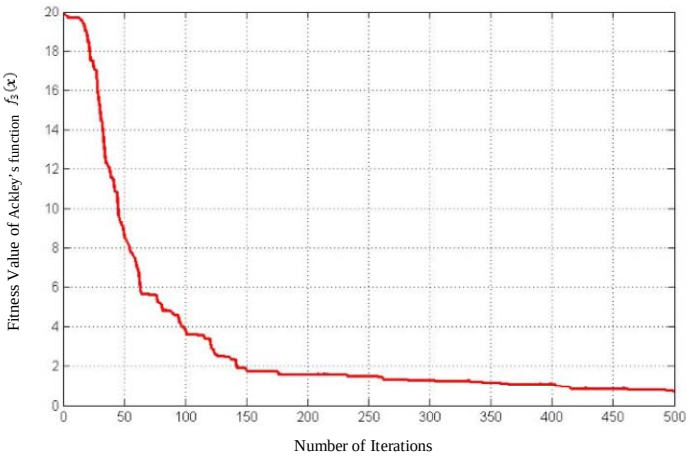

Figure 16. The fitness value of the Ackley's function using PSO.

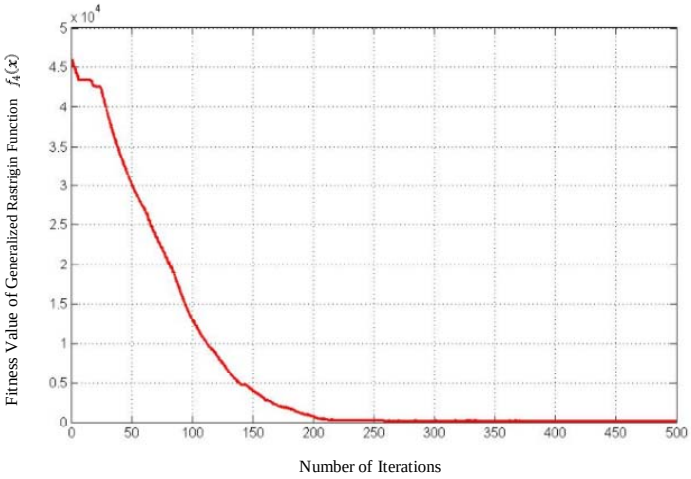

Figure 17. The fitness value of the generalized Rastrigin function using MCPSO-K.

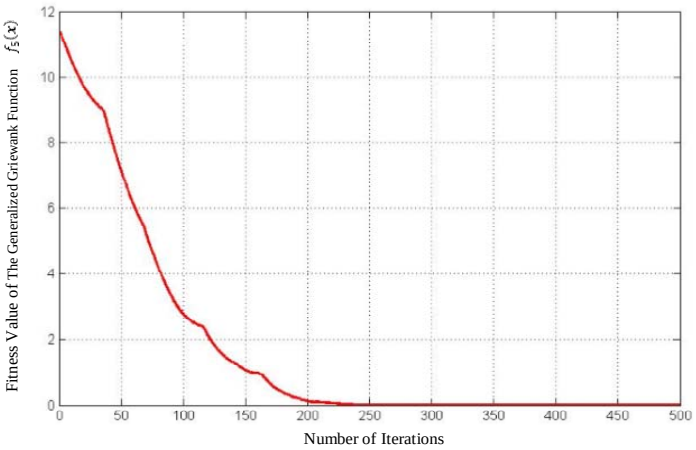

Figure 18. The fitness value of the generalized Griewank function using MCPSO-K.

Table 6. The historical convergence dta using MCPSO-K.

Function	ITERATION		
	50	**150**	**250**
$f_1(\boldsymbol{x})$	$2.20 \times 10^9 \sim 2.30 \times 10^9$	$3.68 \times 10^7 \sim 3.93 \times 10^7$	$1.20 \times 10^5 \sim 1.30 \times 10^5$
$f_2(\boldsymbol{x})$	$4.31 \times 10^4 \sim 4.42 \times 10^4$	$8360 \sim 8482$	$645.3 \sim 655.5$
$f_3(\boldsymbol{x})$	$7.9 \sim 8.5$	$1.523 \sim 1.737$	$1.214 \sim 1.474$
$f_4(\boldsymbol{x})$	$2.86 \times 10^4 \sim 3.01 \times 10^4$	$3860 \sim 3884$	$200.8 \sim 213.5$
$f_5(\boldsymbol{x})$	$6.89 \sim 7.12$	$0.98 \sim 1.04$	$0.004 \sim 0.005$

Table 7. The best final convergence value using PSO.

Function	The Final Convergence Value
$f_1(\boldsymbol{x})$	1346
$f_2(\boldsymbol{x})$	2.138
$f_3(\boldsymbol{x})$	0.7105
$f_4(\boldsymbol{x})$	551.7
$f_5(\boldsymbol{x})$	2.27×10^{-5}

From these data above, the multi-swarm cooperative mechanism obtains a better searching accuracy and search stability. In addition, the cooperative approach introduced here performs better and better as the dimensionality of the problem increases (Figure 19).

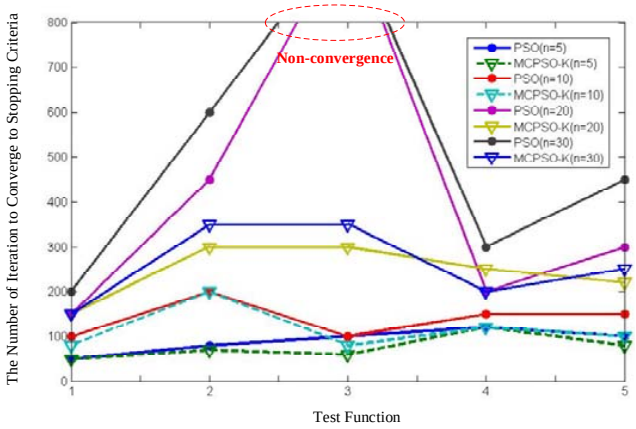

Figure 19. The comparison of algorithm convergence of PSO and MCPSO-K.

5. Conclusions

In this paper, we present an improved multi-swarm cooperative PSO algorithm and compare its performance with the traditional PSO algorithm. The test function results indicate that the MCPSO-K presented in this paper performs better than the traditional PSO in the aspects of convergence, computational efficiency and avoiding premature convergence.

Main differences among the MCPSO-K, PSO and other variants can be summarized as follows:

(1) A new information interaction mechanism is conceived, which is similar to gravitational action mechanism in astrophysics field. In this way, the algorithm can update the velocity of each particle dynamically. In other words, there is an adaptive mechanism to control the searching speed based on the fitness value and the distance of swarms.

(2) Our MCPSO-K adopts an orthogonal initialization method to guarantee the uniform distribution of particles in search space, which avoids local minimum value and the premature convergence.

(3) To greatly decrease the computational cost, a matrix recording the information of particle trajectory is proposed and used during the iteration. By defining a reasonable error range, a group of particles whose variable combination are similar to the particles in the database can be assigned by the value in the matrix directly. Thus, the computational cost by repetitive searches and useless searches is decreased. The test results show that the introduction of the particle trajectory database can decrease the computation cost significantly without influencing the final convergence value.

Acknowledgments: This research is supported by National Nature Science Foundation of China (No. 51575264), the Fundamental Research Funds for the Central Universities (No. NS2015050) and Qing Lan Project.

Author Contributions: Haihua Zhu conceived the concept and performed the research. Jun Yang conducted experiments to evaluate the performance of the proposed algorithm and wrote the manuscript. Yingcong Wang participated in algorithm optimization research and reviewed the manuscript. All authors have read and approved the final manuscript.

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

References

1. Eberhart, R.; Kennedy, J. A new optimizer using particle swarm theory. In Proceedings of the International Symposium on MICRO Machine and Human Science, Nagoya, Japan, 4–6 October 1995.
2. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995.
3. Cheng, Q.; Zhan, C.; Liu, Z.; Zhao, Y.; Gu, P. Sensitivity-based Multidisciplinary Optimal Design of a Hydrostatic Rotary Table with Particle Swarm Optimization. *Strojniski Vestnik* **2015**, *61*, 432–447. [CrossRef]
4. Zouache, D.; Nouioua, F.; Moussaoui, A. Quantum-inspired firefly algorithm with particle swarm optimization for discrete optimization problems. *Soft Comput.* **2016**, *20*, 2781–2799. [CrossRef]
5. Lian, K.L.; Jhang, J.H.; Tian, I.S. A Maximum Power Point Tracking Method Based on Perturb-and-Observe Combined With Particle Swarm Optimization. *IEEE J. Photovolt.* **2014**, *4*, 626–633. [CrossRef]
6. Bonyadi, M.R.; Michalewicz, Z. *Locating Potentially Disjoint Feasible Regions of a Search Space with a Particle Swarm Optimizer*; Springer: New Delhi, India, 2015; pp. 205–230.
7. Abd-Elazim, S.M.; Ali, E.S. A hybrid particle swarm optimization and bacterial foraging for power system stability enhancement. *Complexity* **2015**, *21*, 245–255. [CrossRef]
8. Tran, Q.A.; Quan, D.D.; Jiang, F. Binary Hybrid Particle Swarm Optimization with Wavelet Mutation. In *Knowledge and Systems Engineering*; Springer International Publishing: Cham, Switzerland, 2015; pp. 261–272.
9. Yazdani, D.; Nasiri, B.; Sepas-Moghaddam, A.; Meybodi, M.R. A novel multi-swarm algorithm for optimization in dynamic environments based on particle swarm optimization. *Appl. Soft Comput.* **2013**, *13*, 2144–2158. [CrossRef]
10. Gülcü, Ş.; Kodaz, H. A novel parallel multi-swarm algorithm based on comprehensive learning particle swarm optimization. *Eng. Appl. Artif. Intell.* **2015**, *45*, 33–45. [CrossRef]

11. Dey, S.; Bhattacharyya, S.; Maulik, U. Quantum Behaved Multi-objective PSO and ACO Optimization for Multi-level Thresholding. In Proceedings of the International Conference on Computational Intelligence and Communication Networks, Bhopal, India, 14–16 November 2014.
12. An, Z.Z.; Zhou, H.; Yang, Y.; Shi, X.L. A Study on Particle Trajectory of Particle Swarm Optimization. *Appl. Mech. Mater.* **2013**, *433–435*, 662–666. [CrossRef]
13. Bonyadi, M.; Michalewicz, Z. Stability analysis of the particle swarm optimization without stagnation assumption. *IEEE Trans. Evolut. Comput.* **2016**, *20*, 814–819. [CrossRef]
14. Chen, D.B.; Zhao, C.X. Particle swarm optimization with adaptive population size and its application. *Appl. Soft Comput.* **2009**, *9*, 39–48. [CrossRef]
15. Jin'no, K.; Sano, R.; Saito, T. Particle swarm optimization with switched topology. *Nonlinear Theory Its Appl. Ieice* **2015**, *6*, 181–193. [CrossRef]
16. Ayatollahi, F.; Shokouhi, S.B.; Ayatollahi, A. A new hybrid particle swarm optimization for multimodal brain image registration. *J. Biomed. Sci. Eng.* **2012**, *5*, 18508. [CrossRef]
17. Van den Bergh, F.; Engelbrecht, A.P. A Cooperative approach to particle swarm optimization. *IEEE Trans. Evolut. Comput.* **2004**, *8*, 225–239. [CrossRef]
18. Liang, J.J.; Suganthan, P.N. Dynamic multi-swarm particle swarm optimizer. In Proceedings of the Swarm Intelligence Symposium, Pasadena, CA, USA, 8–10 June 2005.
19. Cheng, S.; Shi, Y. Diversity control in particle swarm optimization. In Proceedings of the Second International Conference on Swarm Intelligence, Chongqing, China, 12–15 June 2011.
20. Trelea, I.C. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Inf. Process. Lett.* **2003**, *85*, 317–325. [CrossRef]
21. Ahila, R.; Sadasivam, V.; Manimala, K. An integrated PSO for parameter determination and feature selection of ELM and its application in classification of power system disturbances. *Appl. Soft Comput.* **2015**, *32*, 23–37. [CrossRef]
22. Cazzaniga, P.; Nobile, M.S.; Besozzi, D. The impact of particles initialization in PSO: Parameter estimation as a case in point. In Proceedings of the IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, Niagara Falls, ON, Canada, 12–15 August 2015.
23. Liang, J.J.; Qin, A.K.; Baskar, S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans. Evolut. Comput.* **2006**, *10*, 281–295. [CrossRef]
24. Branke, J.; Kaußler, T.; Smidt, C.; Schmeck, H. A Multi-population Approach to Dynamic Optimization Problems. In *Evolutionary Design and Manufacture*; Springer: London, UK, 2000; pp. 299–307.
25. Li, C.; Yang, S. Fast Multi-Swarm Optimization for Dynamic Optimization Problems. In Proceedings of the 2008 Fourth International Conference on Natural Computation, ICNC '08, Jinan, China, 18–20 October 2008.
26. Mohais, A.S. Random Dynamic Neighborhood Structures in Particle Swarm Optimization. Ph.D. Thesis, University of the West Indies, Mona, Jamaica, 2008.
27. Huang, T.; Mohan, A.S. Significance of neighborhood topologies for the reconstruction of microwave images using particle swarm optimization. In Proceedings of the 2005 Asia-Pacific Microwave Conference, Suzhou International Conference Center, Suzhou, China, 4–7 December 2005; p. 4.
28. Kennedy, J. Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. In Proceedings of the 1999 Congress on Evolutionary Computation, CEC '99, Washington, DC, USA, 6–9 July 1999.
29. Kennedy, J.; Mendes, R. Population structure and particle swarm performance. In Proceedings of the 2002 Congress on Evolutionary Computation, CEC '02, Honolulu, HI, USA, 12–17 May 2002.
30. Hu, X.; Eberhart, R. Multiobjective optimization using dynamic neighborhood particle swarm optimization. In Proceedings of the 2002 Congress on Evolutionary Computation, CEC '02, Honolulu, HI, USA, 12–17 May 2002.
31. Wang, Q.-H.; Tong, Q.; Han, J.-J.; Bao, B.-Y.-M.Q.-E.; Wu, J.-S.; Han, N.-R.-C.-K.-T.; Dai, N.-Y.-T.; Wu, R.-J. Orthogonal test design for optimization of the isolation and purification of total favonoids from Artemisia frigida Willd using macroporous resin chromatography. *Afr. J. Pharm. Pharmacol.* **2016**, *10*, 192–199. [CrossRef]