ORIGINAL RESEARCH

Comparison of centralized and distributed intelligent particle multi-swarm optimization on search performance

Hiroshi Sho*

Kyushu Institute of Technology, Japan

Received: November 14, 2020 **Accepted:** January 31, 2021 **Online Published:** March 31, 2021
DOI: 10.5430/air.v10n1p1 **URL:** https://doi.org/10.5430/air.v10n1p1

ABSTRACT

In recent years, the technology of particle swarm optimization (PSO) is expanding remarkably. Especially, the technical development of particle multi-swarm optimization (PMSO) attracts attention, and it is expected to handle complex optimization problems. In this paper, we propose two kinds of search methods of PMSO for pattern classification. The crucial idea, here, is how to handle the given parity problems by using these search methods of centralized and distributed intelligent particle multi-swarm optimization (i.e., CIPMSO and DIPMSO). Due to accomplish the hard task of obtaining the high-performance and high-efficiency of PMSO technology, many computer experiments are carried out to handle the 2-bit, 3-bit and 4-bit parity problems under different search situations. Therefore, the obtained experimental results are analyzed and compared, respectively, the search performance and characteristics of the search methods of both CIPMSO and DIPMSO are clarified. Based on the obtained information and know-how, it will further improve the search efficiency and act in conformity of PMSO technology.

Key Words: Evolutionary computation, swarm intelligence, Particle multi-swarm optimization, Parallel processing, Pattern classification, Feedforward neural network

1. INTRODUCTION

In recent years, particle swarm optimization (PSO)[1] technology has been significantly developed[2–5] in the field of evolutionary computation (EC,Evolutionary computation is a family of meta-heuristic optimization algorithms, which involves ant colony optimization, differential evolution, evolutionary programming, genetic algorithms, particle swarm optimization, self-organizing map, and so on.). Particularly, many attentions are paid to the technical development of particle multi-swarm optimization (PMSO).[6–8] Briefly speaking, PMSO is a variant of PSO, which handles a given optimization problem by using multiple particle swarms with parallel processing. The current research situation is to be expected that dealing with some relative complex optimization problems by performing the search methods of PMSO technology.

Under the background of pattern classification, non-linear classible parity problem is a challenging benchmark one for artificial neural networks (ANNs). As a feature of the hard problem, with the growth of the bit number of a given vector, it is difficult to clearly separate the given vectors (patterns) into 0-class or 1-class according to the odd and even of 1's element in each vector.

Taking the advantage of the above search methods of PMSO technology, parallel processing of particle multi-swarm is per-

*Correspondence: Hiroshi Sho; Email: zhang@brain.kyutech.ac.jp; Address: Kyushu Institute of Technology, Japan.

formed with some formations, which have powerful search performance and search efficiency. In this paper, we attempt to handle the given 2-bit, 3-bit and 4-bit parity problems by using two kinds of the search methods of PMSO technology for the first time.

In order to find out the best solution of a given optimization problem, as a search result, not only does it increase the success rate of searching, but also increase the diversity of searching. Since the information about the best solution of each particle swarm is shared with each other, it is possible for PMSO technology to more quickly find out the global best solution of the given problem.

Regarding its weaknesses, there is no guarantee about the optimality of the obtained solution, the search performance depends on the operation way and setting parameters of the search method itself.

Due to overcome and improve the above issues as well as possible, we propose two kinds of the search methods of centralized and distributed intelligent particle multi-swarm optimization (i.e., CIPMSO and DIPMSO).[9] Since parallel processing and intellectual judgment in their mechanisms are performed, the proposed search methods can be confirmed to handle efficiently the given benchmark problems.[10] Also, by repeatedly implementing a particle multi-swarm search within a limited number of searching, the performance effect of dealing with initial stagnation in the search process was increased.

However, since the intellectual judgment way in the search method of CIPMSO is executed by using the evaluative value of the best solution of the particle multi-swarm search, the consistency of each particle swarm in search process is not guaranteed. In contrast, since the intellectual judgment way in the search method of DIPMSO is executed by using the evaluative value of the best solution of each particle swarm search, the consistency of each particle swarm search is guaranteed.

In this paper, two kinds of the search methods of CIPMSO and DIPMSO are described and compared in detail, which be carried out in our computer experiments for handling the given parity problems that are benchmark ones of pattern classification. The purpose of this study is that primarily dealing with relative complex optimization problems for enhancing the search performance and search efficiency of PMSO technology in the application region of pattern classification.. To verify the obtained experimental results by examining and analyzing these experimental data. Thus, the search performance, characteristics, and effects of the proposed search methods will be clarified.

The rest of this paper is organized as follows: Section II introduces the basic search mechanisms and the built-in characteristics of PMSO technology, respectively. Section III provides the experimental results obtained by implementing each search method of both CIPMSO and DIPMSO in our computer experiments, and analyzes the obtained search results for confirming the search performance, capability and characteristics of the used search methods with different combinational modes, respectively. Finally, concluding the obtained remarks, information, and know-how for PMSO technology are given in Section IV.

2. Search methods of CIPMSO and DIPMSO

Due to compare the search performance and characteristics of the search methods of both CIPMSO and DIPMSO, we will introduce the used search methods. For the sake of convenience, let the number of particles in one swarm be N in the D dimensional search space, i.e., $X \in \Re^D$. The position of the ith particle is $\vec{x}^i = (x_1^i, x_2^i, \cdots, x_D^i)^T$ and the velocity of that is $\vec{v}^i = (v_1^i, v_2^i, \cdots, v_D^i)^T$.

2.1 Basic search method of PMSO

Figure 1 shows the flowchart of the proposed two kinds of the search methods of CIPMSO and DIPMSO. It is clear that two mechanisms of the used search methods are quite different, the operation contents of each processing part are almost the same. Each operation part for them is described in detail below.

First, about the “PSO Search" part, it is possible to search and update the incorporated particle swarm optimizer. Specifically, it executes the following search.

Second, in the search process, the update of the particle position of the particle swarm optimizer (the original particle swarm optimizer, the PSO)[1] is expressed by the following equation.

$$\vec{x}_{k+1}^i = \vec{x}_k^i + \vec{v}_{k+1}^i \tag{1}$$

where, the calculation of particle’s velocity $\vec{v}_{k+1}^i$ is given below.

$$\begin{aligned}\vec{v}_{k+1}^i = \; & w_0 \vec{v}_k^i + w_1 \vec{r}_1 \otimes (\vec{p}_k^i - \vec{x}_k^i) + w_2 \vec{r}_2 \otimes (\vec{q}_k - \vec{x}_k^i) \\ & + w_3 \vec{r}_3 \otimes (\vec{s}_k - \vec{x}_k^i)\end{aligned} \tag{2}$$

Eq. (2) is the calculation of the particle’s velocity of the particle multi-swarm search composed of four terms.[7] Here, $\vec{p}_k$ is the best solution found by searching for the particles themselves, $\vec{q}_k$ is the best solution found by searching for particle swarms, $\vec{s}_k$ is the best solution (candidate solution) found by the particle multi-swarm search. $\vec{r}_1$, $\vec{r}_2$ and $\vec{r}_3$ are random vectors with average zero and variance 1, respectively. w_0,

w_1, w_2 and w_3 are setting coefficients. The symbol $\otimes$ is a vector operator.

Third, about the "Renew the best solutions" part, the best solution of the particle itself in the search process p_k^i, the best solution q_k^i of the particle swarm and the best solution s_k of the particle swarm are updated, respectively.

Fourth, about the "Condition Judgment" part, it is to install a sensitive indicator to detect the phenomenon of initial stagnation.[8] Therefore, intellectual judgment can be realized in the search process. The formulation of the sensing indicator is given by the following equation.

$$y_k(L,\varepsilon)=sign\left(\max\left(\varepsilon-\sum_{l=1}^{L}\frac{|g(\vec{s}_k)-g(\vec{s}_{k-l})|}{L},\ 0\right)\right) \quad (3)$$

Here, the $\vec{s}_k$ in Eq. (3) is the best solution for particle multiswarm search at the time k. $g_*(\cdot)$ is the evaluative criterion. This sensitive indicator has two setting parameters, that are, L: measurement data length and ε: measurement standard judgment value. The output value of the sensitive indicator is 1 or 0.

Fifth, about the "Initialization Process" part, if the output of the above sensitive indicator is 1, this situation means that the search should be carried out to initialization process for the related particle swarms.

2.2 Expansion of Bbasic search method of PMSO

Figure 1 shows two general conceptual diagrams for executing two kinds of the search methods of CIPMSO and DIPMSO, respectively, by using the PSO[1] to search. Then if you want to implement other search methods at the "PSO search" part, you just need to replacement the different calculation of the particle's velocity of the particle swarm optimizer.

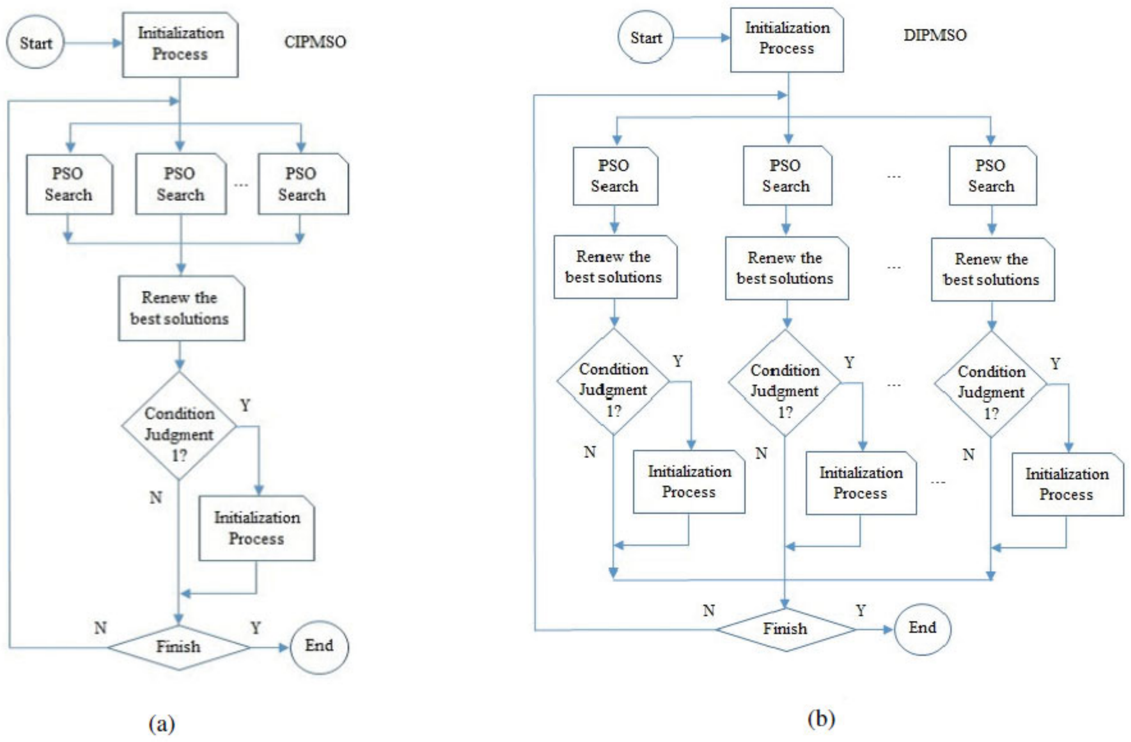

Figure 1. The flowcharts of two kinds of the search methods of CIPMSO and DIPMSO. (a) CIPMSO; (b) DIPMSO.

For example, when executing the particle swarm optimizer with inertia weight (PSOIW) proposed by Shi et al.[11] in the proposed search methods. The calculation of the particle's velocity $\vec{v}_{k+1}^i$ can be changed as follows.

$$\begin{aligned}\vec{v}_{k+1}^i = \ & z(k)\,\vec{v}_k^i+w_1\vec{r}_1\otimes(\vec{p}_k^i-\vec{x}_k^i)+w_2\vec{r}_2\otimes(\vec{q}_k-\vec{x}_k^i)\\ & +w_3\vec{r}_3\otimes(\vec{s}_k-\vec{x}_k^i)\end{aligned} \quad (4)$$

where $z(k)$ is the time-varying linear inertial coefficient. Its value is defined by $z(k) = w_s + (w_e - w_s) \times \frac{k}{K}$ in the search process. Here, K is the number of finish condition for implementing the search methods.

Also, when executing the canonical particle swarm optimizer (CPSO) proposed by Clerc et al.[12] in the proposed search methods, the calculation of the particle's velocity $\vec{v}^{\,i}_{k+1}$ is expressed by the following equation.

$$\begin{aligned}\vec{v}^{\,i}_{k+1} = \ \Phi\Big(&\vec{v}^{\,i}_k + w_1\vec{r}_1 \otimes (\vec{p}^{\,i}_k - \vec{x}^{\,i}_k) + w_2\vec{r}_2 \otimes (\vec{q}_k - \vec{x}^{\,i}_k) \\ &+ w_3\vec{r}_3 \otimes (\vec{s}_k - \vec{x}^{\,i}_k)\Big)\end{aligned} \tag{5}$$

For the sake of convenience, the symbol M1 (the PSO), M2 (PSOIW), and M3 (CPSO) are called for updating and calculating various search methods, i.e., Eq. (2), Eq. (4) and Eq. (5), respectively.

In addition, it should be noted. In using M1 mode, the setting coefficient are $w_0 = 1.0$, $w_1 = w_2 = 2.0$. In using M2 mode, the setting coefficient are $w_s = 0.9$, $w_e = 0.4$, $w_1 = w_2 = 2.0$. In using M3 mode, the setting coefficient are $\Phi = 0.729$, $w_1 = w_2 = 2.05$. And in each mode, the setting coefficient is $w_3 = 0.5$.

In following computer experiments, taking the symbol M123 as an example of the search method to be used. This is a combinational search method for particle multi-swarm optimization, which means that a mixed composition of M1, M2 and M3 to be carried out simultaneously for dealing with a given optimization problem with using M123 combinational mode.

2.3 About the setting evaluative criterion

A feedforward neural network is represented by the particle (individual) of the particle swarm. The connection weights and biases of each particle, which are arranged in a certain order to form one individual (vector).

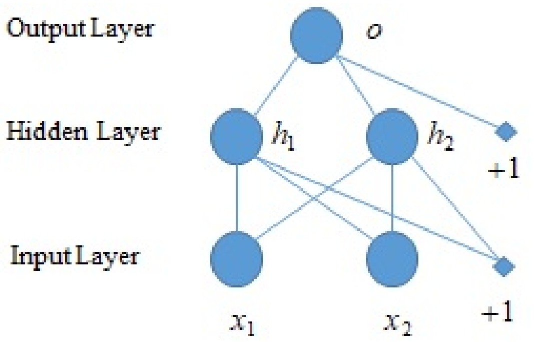

Figure 2. The basic connection structure (2-2-1) of the used three-layer neural network as an evaluative criterion for handling the 2-bit parity problem

About the used feedforward neural network with three-layer is shown in Figure 2, input layer that is constructed by full connection weights between the hidden layer and the output layer. These connection weights and biases are adjustable. It is an estimated target for particle multi-swarm search.

What should be noted here is that the evaluative criterion and the target of estimation are the same. The parameters of the initial evaluative criterion are given at random. As an initial result, the evaluation value of the search results during the initial stage of searching is lower. Afterward, it gradually rises along with the search progresses. Finally, it will settle in a certain stable state with searching.

From the above described, the higher the evaluation value of the evaluative criterion is, the higher the estimation target that approaches the connection weights and biases (i.e., the best solution) of the true feedforward neural network configuration is. However, for distributed representation of connection weights in the feedforward neural network, the composition of the obtained best solution changes significantly.

Since the given parity problem is linearly indistinguishable, when dealing with two kinds of the search methods of CIPMSO and DIPMSO, the output result of the feedforward neural network with non-linearity is used as the evaluative criterion. Specifically, the computational relationship of forward propagation between input and output of the used feedforward neural network is expressed as follows.

$$o(\vec{x}|\vec{w}) = f\left(\sum_{j=1}^{H} w_{kj} f\Big(\sum_{i=1}^{D} w_{ji}x_i + w_{j0}\Big) + w_{k0}\right) \tag{6}$$

where $f(\cdot)$ is a logistics function, that is, $f(z) = \frac{1}{1+e^{-3z}}$. Here, H in the above equation is the number of neuron units in the hidden layer. And w_{ji} and w_{j0} are the connection weights from the hidden layer to the output layer, respectively. Join these connection weights from input layer to hidden layer, also, w_{kj} and w_{k0}, the bias of the neuron unit of output layer and the bias of the neuron unit of hidden layer.

Eq. (7) represents the mean squared error (MSE), which indicates the discriminant effect of the relevant to the used neural feedforward network shown in Figure 2.

$$e(\vec{x}) = \frac{1}{2}\sum_{i=1}^{N} \big(t_i(\vec{x}) - o_i(\vec{x}|\vec{w})\big)^2 \tag{7}$$

where, N is the number of the input/target patterns.

In this case, the evaluative criterion, $g(\cdot)$, is used obviously for the iterative search of both CIPMSO and DIPMSO is given as follows.

$$g(\vec{x}) = \frac{1}{1 + e(\vec{x})} \tag{8}$$

When the MSE, $e(\vec{x})$, of the discriminant effect approaches zero, the evaluative criterion, $g(\vec{x})$, approaches 1.0. Therefore, the given parity problem will be handled with high-accuracy.

3. COMPUTER EXPERIMENTS

Without loss of generility, in the following computer experiments, the given 2-bit, 3-bit and 4-bit parity problems are adopted to facilitate result comparison and analysis of search performance and effects of the proposed two kinds of the search methods of CIPMSO and DIPMSO, respectively. Based on the obtained experimental data, these search results are analyzed for the search performance and search efficiency of each search method.

3.1 Solving 2-bit parity problem (1)

Table 1 shows the input/target patterns for handling the 2-bit parity problem. $\vec{x} = \{x_1, x_2\}$ is the ith input vector, $t_i(\vec{x})$ is the target that corresponds to the ith input vector.

Table 1. Input/target patterns of the 2-bit parity problem

i	x_1	x_2	$t_i(\vec{x})$
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

Here, due to handle the 2-bit parity problem in a particle multi-swarm search, a 3-layer feedforward neural network is adopted as the evaluative criterion. The basic connection structure (2-2-1) of the feedforward neural network is shown in Figure 2. This is a search problem with 9-D in this case.

The given 2-bit parity problem is solved by every search method of CIPMSO and DIPMSO with using M123 combinational mode. As a detailed search condition, the number of particle swarms is 3 for searching in this section, the number of particles in each particle swarm is 10, the setting parameters of the sensitive indicator are L=100 and $\varepsilon = 0.0001$, respectively. The number K is set to 400 for searching.

Under the above experimental conditions, 10 iterative searching are performed for each computational experiment[1], the search results obtained as an example are shown in Figure 3. In either search process, we are able to find out the best solution to the 2-bit parity problem together with both search cases.

From the obtained search results of Figure 3(a)-(a'), the fitness value of the evaluative criterion for the best solution of multiple swarms obtained by the search methods of CIPMSO and DIPMSO are sufficiently close to 1.0. It can be confirmed that the true best solution for handling the given 2-bit parity problem has been found. Also, since the change of the evaluative value, Bx, of the best solution obtained by the search method of DIPMSO remains 1.0 in the search process. This is because the search method is consistent with the search.

Figure 3(b)-(b') show how the fitness values of the best solutions of the used three particle swarms in the multi-swarm search change. The state of the change fully expressed the search characteristics of both CIPMSO and DIPMSO. The symbol B1 is the change in the fitness values of the best solutions found out by No.1 particle swarm, B2 shows the change in the fitness values of the best solutions found out by No.2 particle swarm, and B3 shows the change in the fitness values of the best solutions found out by No.3 particle swarm, respectively.

Figure 3(c)-(c') give the obtained search results, i.e., full connection weights (including biases) of two search methods of CIPMSO and DIPMSO, respectively. The obtained best connection weights of two feedforward neural networks among the many best solutions are shown.

In Figure 3(d)-(d'), the discriminant curved surfaces between the input layer and the hidden layer of both CIPMSO and DIPMSO are shown. Depending on its distribution position, it can be confirmed that the given 2-bit parity problem is completely solved by these discriminant curved surfaces (i.e., And and OR problems).

Figure 3(e)-(e') completely solve the given 2-bit parity problem (XOR problem) of the discriminant curved surfaces between the input layer and the output layer for the search methods of both CIPMSO and DIPMSO can be confirmed. Since there is a large different in the distribution of two discriminant curved surfaces, it reflects the variety of best possible solutions to problem-solving.

Moreover, the success rate of complete discrimination (i.e., the evaluative value of the best solution reaches 1.0) is 100%. Therefore, it is shown that the search performance and effect of the search methods of both CIPMSO and DIPMSO are very high and well.

[1] Computational environment–Hardware: Dell Optiplex 3020, Intel (R) core (TM) i5-4590, CPU 3.3GHz, Ram 8.0GB; Computational Tools–Software: Mathematica 11.3; Calculation time: Approximately 16 sec.

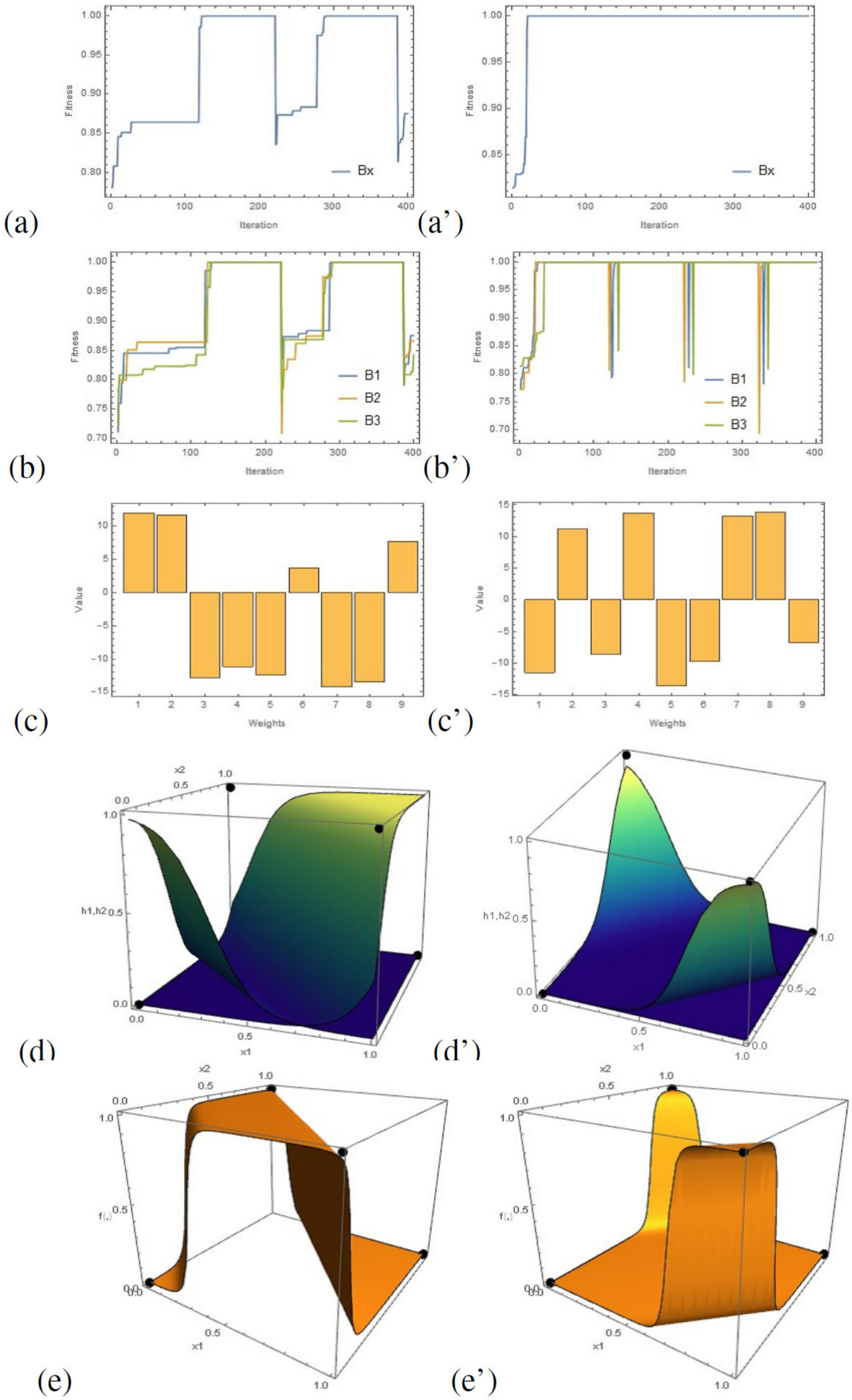

Figure 3. The obtained experimental results of implementing the search methods of both CIPMSO and DIPMSOD (a)-(a') Change in the fitness value of the best solution of the particle multi-swarm during the search process; (b)-(b') Change in the fitness value of the best solution of each particle swarm search during the search process; (c)-(c') Vector configuration of the best solution; (d)-(d') Discriminat curved surface of patterns in the hidden layer; (e)-(e') Discriminat curved surface of patterns in the output layer.

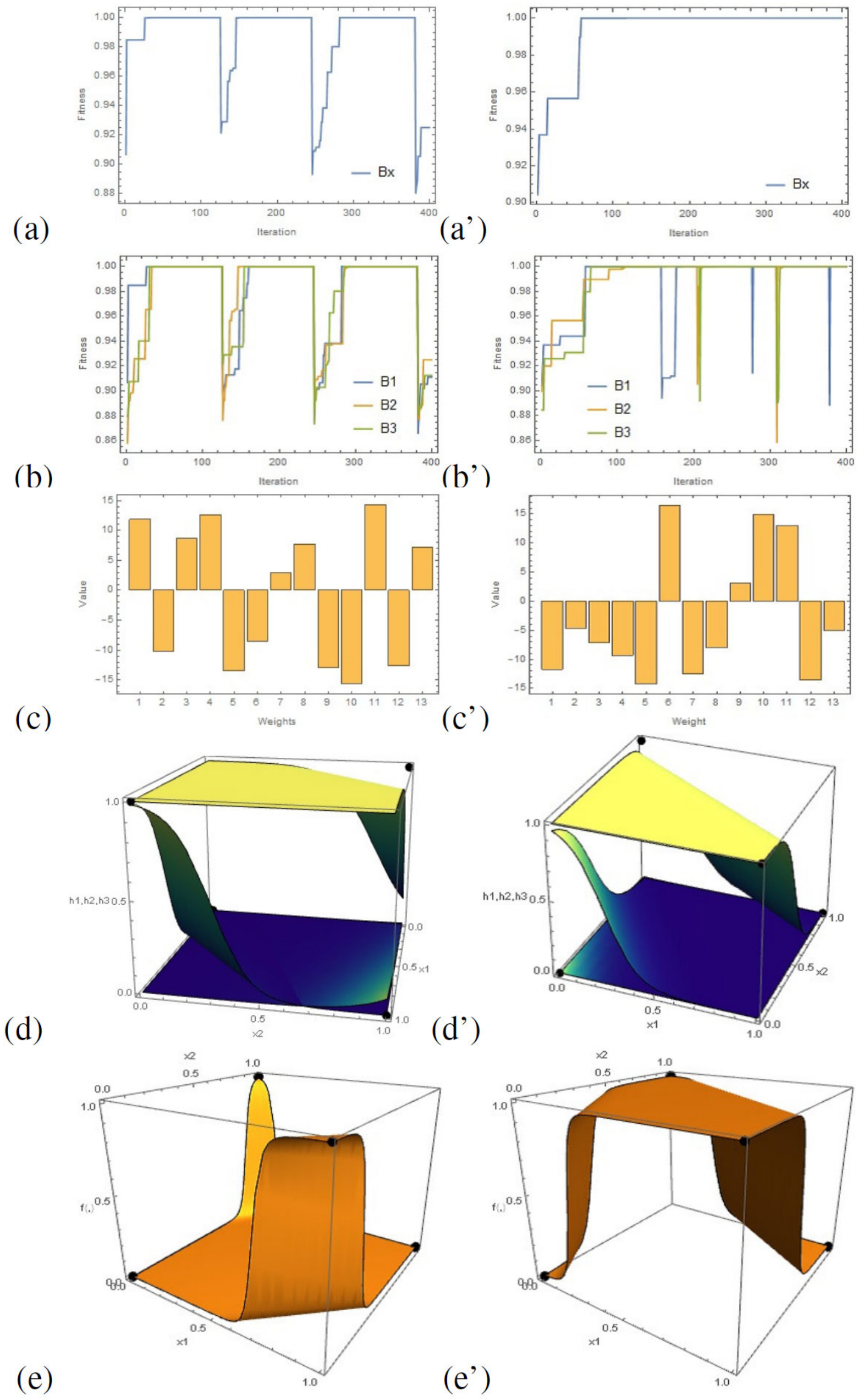

Figure 4. The obtained experimental results of implementing the search methods of both CIPMSO and DIPMSOD (a)-(a') Change in the fitness value of the best solution of the particle multi-swarm during the search process; (b)-(b') Change in the fitness value of the best solution of each particle swarm search during the search process; (c)-(c') Vector configuration of the best solution; (d)-(d') Discriminat curved surface of patterns in the hidden layer; (e)-(e') Discriminat curved surface of patterns in the output layer.

3.2 Solving the 2-bit parity problem (2)

As same as the previous section, we continually handle the 2-bit parity problem in this section. But the number of neuron units in the hidden layer is set to three instead of two. Therefore, the basic connection structure (2-3-1) of the feedforward neural network is used. So this is a search problem with 13-D in this case..

By adding a redundant neuron unit to analyze the search situation and observe how about the particle multi-swarm search plays a role in pattern classification. As an example, the obtained results of computer experiments by implementing each search methods of CIPMSO and DIPMSO with using M123 combinational mode are shown in Figure 4.

From the obtained experimental results of Figure 4(a)-(a'), we can see that the search methods of both CIPMSO and DIPMSO are found to have higher success rate for searching. The search features and performance effects of two kinds of the search methods of CIPMSO and DIPMSO can be confirmed.

From the obtained experimental results shown in Figure 4(b)-(b'), the best search behavior of each particle swarm can be observed.

From the obtained experimental results shown in Figure 4(c)-(c'), it is clear that the vector configurations of the best solutions are quite different.

From the obtained experimental results shown in Figure 4(d)-(d'), it can be confirmed that the redundant discriminat curved surfaces (i.e., AND and OR problems) are generated in each case. This means that there are many solutions for completely handling the given problem.

As the final search result, Figure 4(e)-(e') give the discriminat curved surfaces of handling the 2-bit parity problem. We can see that the obtained discriminat curved surfaces clearly separated the 2-bit parity problem with high-accuracy.

3.3 Solving the 3-bit parity problem

In this section, we implement the different search methods of CIPMSO and DIPMSO with using M111, M222, M333, and M123 combinational modes, respectively, to handle the 3-bit parity problem for investigating the search performance and characteristics.

Table 2 shows the input/target patterns for handling the 3-bit parity problem. Regarding the search conditions in the following experiments, it is the same as the used design parameters set in Section 3.1.

Table 2. Input/target patterns of the 3-bit parity problem

i	x_1	x_2	x_3	$t_i(\vec{x})$
1	0	0	0	0
2	0	0	1	1
3	0	1	0	1
4	0	1	1	0
5	1	0	0	1
6	1	0	1	0
7	1	1	0	0
8	1	1	1	1

However, three neuron units are set in the hidden layer of the used feedforward neural network as the evaluative function for handling the 3-bit parity problem. As an evaluative criterion, the basic connection structure (3-3-1) of the feedforward neural network shown in Figure 4 is used. Thus, this is a search problem with 16-D in this case.

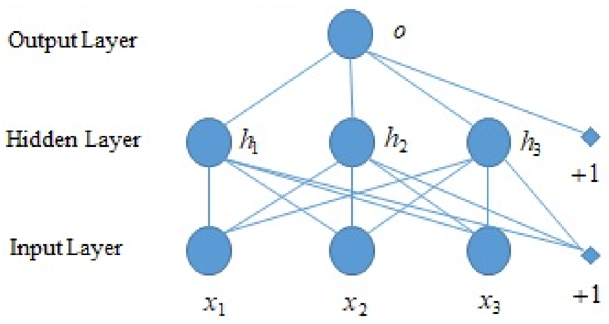

Figure 5. The basic connection structure (3-3-1) of the used three-layer feedforward neural network as an evaluative criterion for handling the 3-bit parity problem

Consequently using two kinds of search methods of CIPMSO and DIPMSO, each repeated search of every method is performed with 10 times to find out the best solution with using M111, M222, M333, and M123 combinational modes, respectively.

For comparison of the search performance, the obtained search results of the different combinational search methods of CIPMSO and DIPMSO are shown in Figure 6. Based on the obtained search results, the search performance of the search methods of CIPMSO and DIPMSO with using M333 combinational mode, are not well among the used other combinational modes. The obtained experimental results suggest that the search performance of the local search method is so lower in despite of using particle multi-swarm way to search.

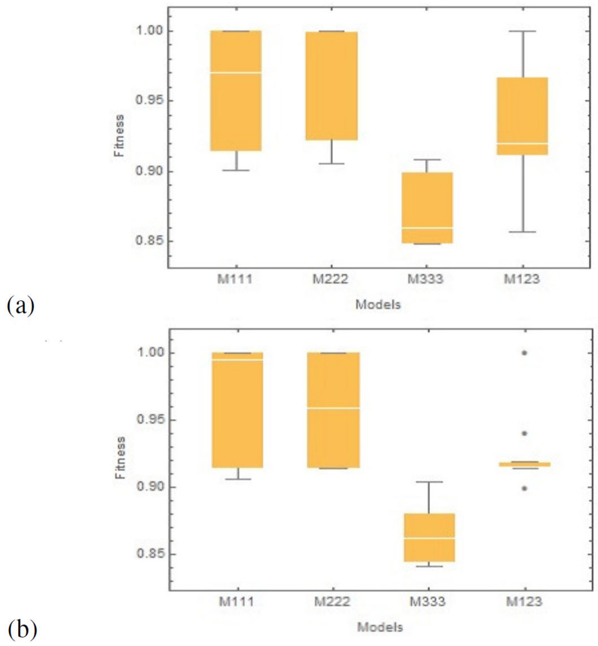

Figure 6. The obtained experimental results of the search methods of both CIPMSO and DIPMSO by implementing different search models. (a) Every search method of CIPMSO; (b) Every search method of DIPMSO.

On the other hand, not limited to this result, it can be confirmed in other search cases. In contrast to the above search result, the global search or asymptotic local search method that gives better search performance.

In this case, the success rates of the complete discrimination between two search methods of CIPMSO and DIPMSO are quite high, respectively. Based on the above search results except using M333 combinational mode, for the information on how to search by implementing two search methods, it is found that there is not so big different in the search performance and search efficiency between two search methods even with pattern classification.

3.4 Solving the 4-bit parity problem

In this section, we implement the different search methods of both CIPMSO and DIPMSO with using M111, M222, M333, and M123 combinational modes, respectively, to handle the 4-bit parity problem for investigating the search performance and characteristics.

Table 3 shows the input/target patterns for handling the 4-bit parity problem. Regarding the search conditions in the following experiments, it is the same as the used design parameters set in Section 3.1.

As an evaluative criterion for handling the 4-bit parity problem, the basic connection structure (4-4-1) of the feedforward neural network shown in Figure 7 is used. Therefore, this is a search problem with 25-D in this case.

Table 3. Input/target patterns of the 4-bit parity problem

i	x_1	x_2	x_3	x_4	$t_i(\vec{x})$
1	0	0	0	0	0
2	0	0	0	1	1
3	0	0	1	0	1
4	0	0	1	1	0
5	0	1	0	0	1
6	0	1	0	1	0
7	0	1	1	0	0
8	0	1	1	1	1
9	1	0	0	0	1
10	1	0	0	0	0
11	1	0	1	0	0
12	1	0	1	1	1
13	1	1	0	0	0
14	1	1	0	1	1
15	1	1	1	0	1
16	1	1	1	1	0

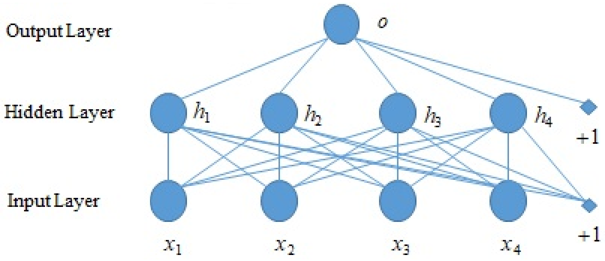

Figure 7. The basic connection structure (4-4-1) of the used three-layer feedforward neural network as an evaluative criterion for handling the 4-bit parity problem

Implementing the different search methods of CIPMSO and DIPMSO with using M111, M222, M333, and M123 combinational modes, respectively, to handle the 4-bit parity problem. Each repeated search is performed with 10 times to find out the best solution. For comparison of search performance and search efficiency, the obtained search results are shown in Figure 8.

We can see that the success rates of the complete discrimination between two search methods are zero, respectively, among the obtained search results. There are two reasons for the obtained better search results. One is that the 4-bit parity problem is so hard one for completely handling it. Another one is that the given search condition is needed to change, e.g., increase the repeated number for searching, change the two setting parameters of the sensitive indicator, and so on.

Based on the above obtained search results, for information on how to handle the given parity problem by implementing two search methods of CIPMSO and DIPMSO, it is found that there is no big different in the search performance between two search methods even with pattern classification.

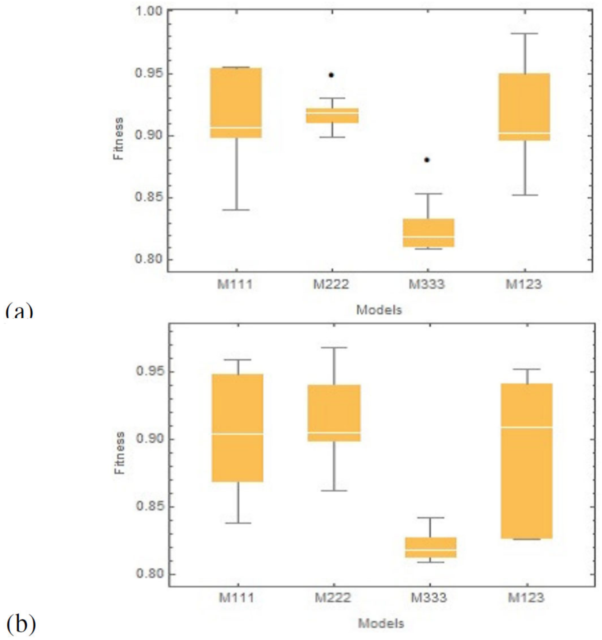

Figure 8. The obtained experimental results of the search methods of both CIPMSO and DIPMSO by implementing different search models. (a) Every search method of CIPMSO; (b) Every search method of DIPMSO.

4. Conclutions

In this paper, we proposed the new search methods to handle the given 2-bit, 3-bit and 4-bit parity problems of pattern classification, which use two kinds of the search methods of centralized and distributed intelligent particle multi-swarm optimization (CIPMSO and DIPMSO). This is the first endeavor to handle the relative complex optimization problems on pattern classification for enhancing PMSO technology. Through many computer experiments, the 2-bit parity problem was handled and analyzed. Furthermore, we also successfully handled the 3-bit and 4-bit parity problems by the proposed search methods of them.

And by the growth of the bit number of the given parity problem and adding a redundant neuron unit to the hidden layer of the feedforward neural network, based on the large amount of experimental data obtained. The performance effects of two solutions and their changing characteristics were clarified. Based on the experimental results of these pattern classifications, the search features, performance effects, and effectiveness of two kinds of the search methods of CIPMSO and DIPMSO were reconfirmed.

Based on using the feedforward neural network to represent the evaluative criterion, the range of applications for particle multi-swarm optimization has been greatly expanded. This is a new step toward technological development and its application related to particle multi-swarm optimization itself.

As a topics of the feature research, a hybrid search method which including the mechanism of differential evolution (DE)[9] can be constructed for the search methods of the centralized and distributed intelligent particle multi-swarm optimization. Taking advantage of the introduction of different search mechanisms, it is expected that more better search performance can be obtained in near future study.

References

[1] Kennedy J, Eberhart RC. Particle swarm optimization. Proceedings of the 1995 IEEE International Conference on Neural Networks. 1995; 1942-1948. Perth, Australia.

[2] Aote SA, Raghuwanshi MM, Malik L. A Brielf Review on Particle Swarm Optimization: Limitations & Future Directions. International Journal of Computer Science Engineering. 2013; 2(5): 196-200.

[3] El-Abd M, Kamel MS. A Taxonomy of Cooperative Particle Swarm Optimizers. International Journal of Computational Intelligence Research. 2008; 4(2): 137-144.

[4] Sundaray M, Tripathy AK, Tripathy SK. A new algorithm based on particle swarm optimization for application in holographic coupler. 2020; *Optik*, 208: Article 164551. `http://doi.org/10.16/j.ijleo.2020.164551`

[5] Xia X, Gu L, He G, et al. An expanded particle swarm optimization based on multi-exemplar and forgetting ability. Information Sciences. 2020; (508): 105-120.

[6] Niu B, Zhu Y, He X, et al. Multi-population Cooperation Particle Swarm Optimization. 2005; LNCS 3630, 874-883. Springer Heidelberg. `http://doi.org/10.1007/11553090${}_{-}^{}$88`

[7] Sho H. Particle Multi-Swarm Optimization: A Proposal of Multiple Particle Swarm Optimizers with Information Sharing. Proceedings of 2017 10th International Workshop on Computational Intelligence and Applications. 2017; 109-114. Hiroshima, Japan.

[8] Zhang H. Multiple Particle Swarm Optimizers with Diversive Curiosity. Proceedings of the IAENG International MultiConference of Engineers and Computer Scientists. 2010; 1: 174-179. Hong Kong, China.

[9] Sho H. Investigation of Differential Evolution and Particle Swarm Optimization in Search Performance. Artificial Intelligence Research. 2020; 9(1): 54-66. Available from: http://www.sciedupress.com /issues_v28/ issue_3/index.html

[10] Suganthan PN, Hansen N, Liang JJ, et al. Problem Definitions and Evaluation Criteria for the CEC 2005. Available from: http//:www.ntu.edu.sg/home/epnsugan/index_files/CEC-05/Tech-Report-May-30-05.pdf

[11] Shi Y, Eberhart RC. A modified particle swarm optimiser. Proceedings of the IEEE International Conference on Evolutionary Computation. 1998; 69-73. Anchorage, Alaska, USA.

[12] Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation. 2000; 6(1): 58-73.

[13] Duda RO, Hart PE, Stock DG. Pattern Classification, John Wiley & Sons Inc, USA; 2001.

[14] Sho H. Investigation of Particle Multi-Swarm Optimization with Diversive Curiosity. Engineering Letters. 2020; 28(3): 960-969. Available from: http://www.engineeringletters.com/issues_v28/issue_3/index.html