ORIGINAL RESEARCH

Investigation of differential evolution and particle swarm optimization in search performance

Hiroshi Sho*

Kyushu Institute of Technology, Japan

Received: August 18, 2020 **Accepted:** October 19, 2020 **Online Published:** October 31, 2020
DOI: 10.5430/air.v9n1p54 **URL:** https://doi.org/10.5430/air.v9n1p54

ABSTRACT

The purpose of this study is to clarify the search performance of differential evolution (DE) and particle swarm optimization (PSO) technologies for instinctively understanding the specificity of the used search methods. Due to achieve the task, here, the several search methods of both, i.e., DE/rand/1, DE/rand/2, DE/best/1, DE/best/2, the PSO, PSOIW, and CPSO, which are systematically implemented in this paper. Therefore, many computer experiments are carried out for handling the given five benchmark problems. Through the analysis of the obtained experimental data, the detail search performance and characteristics of them are observed and compared, respectively. From the obtained results, it is found that the search methods of DE/best/1 and the PSO relatively have better search performance. Based on the findings and know-how, they can provide some reference and key hint, i.e., the elitism strategy plays an important role in their stochastic population search approaches, for encouraging development and improvement of both DE and PSO technologies in the near future. And as the applicative examples, the PSO is used to handle typical 2-bit and 3-bit parity problems for the pattern classification.

Key Words: Evolutionary computation, Swarm intelligence, Differential evolution, Particle swarm optimization, Pattern classification, Parity problem

1. INTRODUCTION

In the field of evolutionary computation (EC)[1], differential evolution (DE)[1] and particle swarm optimization (PSO)[2,3] are two powerful search approaches to handle many difficult optimization problems, which have a real-valued decision variable. The basic characteristics of both DE and PSO technologies are described below: The former is that there are few setting parameters for creating individual generation, despite the extremely simple search mechanism, which has better search performance. The latter is that by exchanging search information between particles and internal memory, it also has higher search capability. Since the two approaches have directional search feature, even in dealing with a certain complicated optimization problem, the best solution could be found out easily than other stochastic search methods yet.

Based on the common advantages and characteristics of both DE and PSO technologies, in recent year they are actively developed and researched. Therefore, many improved search methods and the obtained search results have been reported and published.[4–6] These search methods do not require the constraint of difficulty or convexity to the evaluative criteria. With the strong reason, they could be widely applied in various different regions, i.e., construction design, route search, traffic management, financial economy, control systems and

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

[1] Evolutionary computation is a family of meta-heuristic optimization algorithms, which involves ant colony optimization, differential evolution, evolutionary programming, genetic algorithms, evolution strategy, particle swarm optimization, self-organizing map, and so on.

so on. Especially, the most researches and projects of both DE and PSO technologies, which focus on the treatment of actual application problems.[7–9]

However, the common weaknesses of both DE and PSO technologies are very similar to other stochastic population search approaches, e.g., genetic algorithms (GA)[10] and evolution strategy (ES)[11] etc., there is no any guarantee about the optimality of the obtained solutions, the search performance depends on the operational ways and setting parameters that they have.

As a relevant study, there was a literature[12] for comparing three evolutionary algorithms: GA, PSO, and DE. Its content mainly focused on the general observations on the similarities and differences among the three algorithms based on computational step were discussed, and qualitative comparison of the three-way for handling job shop scheduling problems was practiced. It also pointed out that the comparisons are often made indirectly since many researchers applied different solution representations in combination with various local search. And a fair comparison of the search performance of different methods should be more comprehensive when the given benchmark problems are used with same solution representation and the same number of evolution.

In contrast to the above prior research and situation, the purpose of this study is to clarify the search performance of DE and PSO technologies for instinctively understanding the specificity of the used search methods.

Due to achieve the task, here, we systematically investigate and compare the search performance and characteristics of their search methods by handling the given five benchmark problems with same number of evolution. And the most basic search methods which are DE/rand/1[1] and the PSO[3] are briefly described and examined by the mechanism of execution, respectively. Then, by implementing each search method, i.e., DE/rand/1, DE/rand/2, DE/best/1, DE/best/2,[4] the PSO, PSOIW,[13–15] and CPSO[?, ?] of both DE and PSO, massive search results could be obtained from our computer experiments. Thus, the detail search performance and characteristics of various search methods of DE and PSO are confirmed, respectively.

Concretely, it is obvious that these findings and know-how which can provide some reference and key hint, i.e., the elitism strategy plays an important role in their stochastic population search approaches, for encouraging development and improvement of both DE and PSO technologies in the near future. Furthermore, it is also expected that creating several hybrid search methods of particle multi-swarm (or individual multi-population) search by parallel process, intelligent judgment, and information sharing with each other,[18, 19] which have more higher search performance and effect to deal with the complicated optimization problems such as pattern classification (PC),[20] job shop scheduling (JSS),[21] design of artificial neural networks (ANNs),[22] multi-objective optimization (MOO),[23] and so on. And as applicative examples, the search method of PSO is used to handle typical 2-bit and 3-bit parity problems for the pattern classification.[24]

The rest of this paper is organized as follows: Section II introduces the basic search mechanisms and the built-in characteristics of both DE and PSO, respectively. Section 3 provides the experimental results obtained by implementing each search method in our computer experiments. Section IV analyzes these obtained search results for confirming the search capabilities and characteristics of the used search methods, respectively. Section V shows that the 2-bit and 3-bit parity problems are carried out by the PSO as the given applicative examples. Finally, the concluding remarks are given in Section VI.

2. Basic search mechanisms of DE and PSO

It is considered, here, that using each search method of both DE and PSO for handling the following non-constraint optimization problem.

$$Min\{f(\vec{x})|\vec{x} \in X\} \tag{1}$$

where, $f(\cdot)$ is a given criterion function, $\vec{x}$ is a real-valued vector (called candidate solution), $X = \Re^D$ is a given search space, D is the number of dimension of the vector $\vec{x}$.

2.1 Basic search mechanism of DE

The mechanism of DE/rand/1 is simply described, which is the most basic search method of DE technology. A collection of N individuals is called as an individual population P_k. The population P_1 is randomly generated according to an uniform distribution. The generation (evolution) process from the individual population P_k to the individual population P_{k+1} is created as follows.

Firstly, a candidate solution $\vec{x}^i \in P_k$ is selected one by one in order, which be called as the target vector.

Secondly, randomly choose the candidate solution $\vec{x}^{r1}$ (base vector), $\vec{x}^{r2}$ and $\vec{x}^{r3}$ from the population P_k. Note: the selected vector $\vec{x}$ should be satisfied with the condition, i.e., $i \neq r1 \neq r2 \neq r3$.

Afterward, the given a random coefficient F, a mutation vector $\vec{v}^i$ is generated from the above three candidate solutions by Eq. (2).

$$\vec{v}^i = \vec{x}^{r1} + F\cdot(\vec{x}^{r2} - \vec{x}^{r3}) \tag{2}$$

The above equation is also called as a mutation operator in DE/rand/1. Continually based on the crossing rate C_r and the randomly selected subscript $j_r \in [1, D]$, an element of $\vec{x}^i_j$ or $\vec{v}^i_j$ as in Eq. (3) of performing crossover operator.

$$\vec{u}^i_j = \begin{cases} \vec{v}^i_j & if(rand^i \leq C_r \vee (j = j_r) \\ \vec{x}^i_j & otherwise \end{cases} \tag{3}$$

Finally, compare the evaluative values of the generated trial vector $\vec{u}^i$ and the target vector $\vec{x}^i$, as shown in Eq. (4), the i-th candidate solution $\vec{z}^i \in X$ of the individual population P_{k+1}.

$$\vec{z}^i = \begin{cases} \vec{u}^i & if\ f(\vec{u}^i) < f(\vec{x}^i) \\ \vec{x}^i & otherwise \end{cases} \tag{4}$$

The above-mentioned operations are repeated, until the end of search, the specified number of search iterations K, is reached.

In addition, the setting parameters of DE/rand/1 are population size N, random coefficient $F(\in (0, 2))$, and the rate of crossover is C_r, respectively.

2.2 Basic search mechanism of PSO

The stochastic search method of PSO was proposed by Kennedy and Eberhart in 1995. Its mechanism for searching is very simple to effectively deal with a given optimization problem. Here, the search method is referred to as the PSO (i.e., the original particle swarm optimizer).

Specifically, in beginning of particle swarm search, position (i.e., candidate solution) and velocity (i.e., amount of change for finding out the best solution) of the i-th particle in the particle swarm are generated at random, then they are updated continuously as follows:

$$\vec{x}^i_{k+1} = \vec{x}^i_k + \vec{v}^i_{k+1} \tag{5}$$

$$\vec{v}^i_{k+1} = w_0\vec{v}^i_k + w_1\vec{r}_1 \otimes (\vec{p}^i_k - \vec{x}^i_k) + w_2\vec{r}_2 \otimes (\vec{q}_k - \vec{x}^i_k) \tag{6}$$

where $\vec{x}^i_k$ refers to the candidate solution of the i-th particle in the given search space X and $\vec{v}^i_k$ refers to its velocity at iteration k, respectively. w_0 is an inertia weight, w_1 is a coefficient for individual confidence, w_2 is a coefficient for swarm confidence. $\vec{r}_1$, $\vec{r}_2 \in \Re^D$ are two random vectors in which each element is uniformly distributed over the range $[0, 1]$, and the symbol $\otimes$ is an element-wise operator for vector multiplication. $\vec{p}^i_k(=arg \underset{j=1,\cdots,k}{max}\{g(\vec{x}^i_j)\}$, where $g(\cdot)$ is the criterion value of the i-th particle at iteration k.) is the local best solution of the i-th particle up to now, and $\vec{q}_k(=arg \underset{i=1,2,\cdots}{max}\{g(\vec{p}^i_k)\})$ is the global best solution among whole particle swarm.

Like the above-mentioned search method DE/rand/1, the setting parameters of the PSO are shown below. $w_0 = 1.0$, w_1 = w_2 = 2.0, $\vec{r}_1$ = $\vec{r}_2 \in (0, 1)^D$, and N is the size of the used particle swarm used.

2.3 Improved search methods of DE/rand/1 and the PSO

For obtaining better search performance and effect of both DE and PSO technologies, based on the above-mentioned basic search mechanisms of DE/rand/1 and the PSO, various improvements and suggestions about them have been published and used.[25,26]

2.3.1 *Improved search methods of DE/rand/1*

Regarding the improved search methods of DE/rand/1, focusing on diversity of individuals, expansion of fluctuation range and information sharing, it is developed for obtaining efficiency to explore. Specifically, the mutation vector $\vec{v}^i$ in Eq. (2) and the trial vector $\vec{u}^i$ in Eq. (3), users can change the generation mode according to some ideas.

There are several improving search methods of DE, i.e., DE/rand/2, DE/best/1, and DE/best/2 etc. of DE,[1] to be created below. For the sake of convenience, the mutation vector $\vec{v}^i$ is generated in DE/rand/2 is given as follows:

$$\vec{v}^i = \vec{x}^{r1} + F\cdot(\vec{x}^{r2} - \vec{x}^{r3}) + F\cdot(\vec{x}^{r4} - \vec{x}^{r5}) \tag{7}$$

where the selected vector $\vec{x}$ should be satisfied with the condition, i.e., $i \neq r1 \neq r2 \neq r3 \neq r4 \neq r5$.

Then, similar to the above, the mutation vector $\vec{v}^i$ is generated in DE/best/1 is given as follows:

$$\vec{v}^i = \vec{x}^{best} + F\cdot(\vec{x}^{r2} - \vec{x}^{r3}) \tag{8}$$

where $\vec{x}^{best}$ is the best solution among whole individual population until now.

And the mutation vector $\vec{v}^i$ is generated in DE/best/2 is given as follows:

$$\vec{v}^i = \vec{x}^{best} + F\cdot(\vec{x}^{r2} - \vec{x}^{r3}) + F\cdot(\vec{x}^{r4} - \vec{x}^{r5}) \tag{9}$$

where the vector $\vec{x}$ should be satisfied with the condition, i.e. $i \neq r2 \neq r3 \neq r4 \neq r5$.

2.3.2 *Improved search methods of the PSO*

Meanwhile, regarding the improved search methods of the PSO, focusing on the accuracy and convergence of search results, it is developed for better stability in search process. For example, there are some search methods such as PSOIW (i.e., particle swarm optimizer with inertia weight) and CPSO (i.e., canonical particle swarm optimizer) of PSO to be created.

For improving the convergence and search ability of the PSO, Shi & Eberhart modified the updating rule of the particle's velocity shown in Eq.(10) by constant reduction of the inertia coefficient over time-step[23, 24] as follows:

$$\vec{v}^{\,i}_{k+1} = w(k)\,\vec{v}^{\,i}_k + w_1\vec{r}_1 \otimes (\vec{p}^{\,i}_k - \vec{x}^{\,i}_k) + w_2\vec{r}_2 \otimes (\vec{q}_k - \vec{x}^{\,i}_k) \quad (10)$$

where $w(k)$ is a variable inertia weights that is linearly reduced coefficient from a starting value, w_s, to a terminal value, w_e, with the increment of time-step k as follows:

$$w(k) = w_s + \frac{w_e - w_s}{K}\,k \quad (11)$$

where K is the maximum number of time-step for the PSOIW searching. In PSOIW, the boundary values are adopted to $w_s = 0.9$ and $w_e = 0.4$, respectively, and $w_1 = w_2 = 2.0$ are still used as in the PSO.

For the same purpose as the above described, Clerc & Kennedy modified the updating rule for the particle's velocity in Eq. (12) by a constant inertia coefficient over time-step[16, 17] as follows:

$$\vec{v}^{\,i}_{k+1} = \Phi\left(\vec{v}^{\,i}_k + w_1\vec{r}_1 \otimes (\vec{p}^{\,i}_k - \vec{x}^{\,i}_k) + w_2\vec{r}_2 \otimes (\vec{q}_k - \vec{x}^{\,i}_k)\right) \quad (12)$$

where Φ is an inertia coefficient relative to w_0. In CPSO, $\Phi = 0.729$, $w_1 = w_2 = 2.05$ are used.

3. Computer experiments

Without loss of generality, in the following computer experiments, a suite of benchmark problems[27] (including three unimodal and two multimodal problems) in 5 or 10 dimensions are adopted to facilitate data comparison and analysis of search performance index of the used search methods. These evaluative criteria to calculate the fitness of the obtained solutions are shown in Table 1, and the search range of all simulation cases is $X \in (-5.12, 5.12)^D$.

The evaluative criterion of the given five benchmark problems i.e., finding the global solution of the given five functions is expressed as follows:

$$g_\star(\vec{x}_k) = \frac{1}{1+f_\star(\vec{x}_k)} \quad (13)$$

where the symbol $\star$ denotes each given benchmark problem, i.e., *Sphere* (Sp), *Griewank* (Gr), *Rastrigin* (Ra), *Rosenbrock (Ro), and Schwefel (Sc) problems,* respectively.

According to the definition of this criterion, it is clear that when the most best solution $\vec{x}^{\,b}_k$ is found out by particle swarm search or individual population search, the output result of the given function $f_\star(\vec{x}^{\,b}_k)$ approaches 0. Therefore, the output value of the evaluative criterion $g_\star(\vec{x}^{\,b}_k)$ will approach 1.

Table 2 shows the major parameters used to explore for dealing with the given optimization problems by implementing the search methods, respectively.

3.1 Comparison (1) of search performance in 5D case

Two basic search methods, i.e., DE/rand/1 and the PSO, are performed for dealing with $Sphere$ problem in 5D case to observed their search performance[2]. By comparison of the obtained search results, the search characteristics of them are investigated.

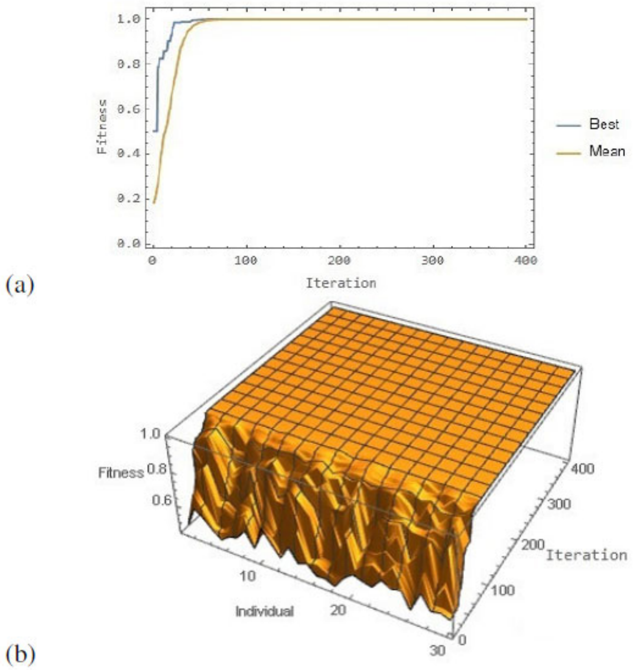


Figure 1. Search result of performing DE/rand/1 for Sphere problem in 5D case. (a) The change of the best and average value in the search process; (b) The change of fitness value of each individual.

As an example, the obtained search result of DE/rand/1 is shown in Figure 1. We can see that the search of DE/rand/1 is carried out smoothly, and the best solution of the given

[2] Computing environment – hardware: DELL OPTIPLEX 3020, Intel(R) core(TM) i5-4590, CPU 3.3GHz, RAM 8.0GB; Software: Mathematica 11.3; Search time: about 1.45 sec.

$Sphere$ problem is found out by the change state of the evaluative values of the best solution and average solution shown in Figure 1(a). Figure 2(b) indicates clearly that the change of the evaluative value of each individual during the whole search process. As the characteristic of DE/rand/1, it can be seen that the search fluctuates slowly to attain the best solution of $Sphere$ problem.

Consequently, the obtained search result of the PSO is shown in Figure 2 for dealing with the same problem in 5D case.

Comparing the obtained search results of both Figure 1(a) and Figure 2(a), it is clear that the PSO can find out the best solution quickly than DE/rand/1 does. Almost the same behavior can be confirmed with respect to changes in the evaluative value of the average solutions of both.

Comparing the obtained search results of both Figure 1(b) and Figure 2(b), it turns out that the convergent situation of the two search methods is almost same. However, it can be seen that the PSO changes slowly more coarse than DE/rand/1 does at the initial stage of whole search process.

The given five benchmark problems shown in Table 1 are executed (10 trials for each case) by using every search method of both DE and PSO, which is introduced in Section 2. As the statistical data of implementation, the obtained search results, i.e., the maximum value, average value, and standard deviation (S.D.) are shown in Table 3 for dealing with each benchmark problem.

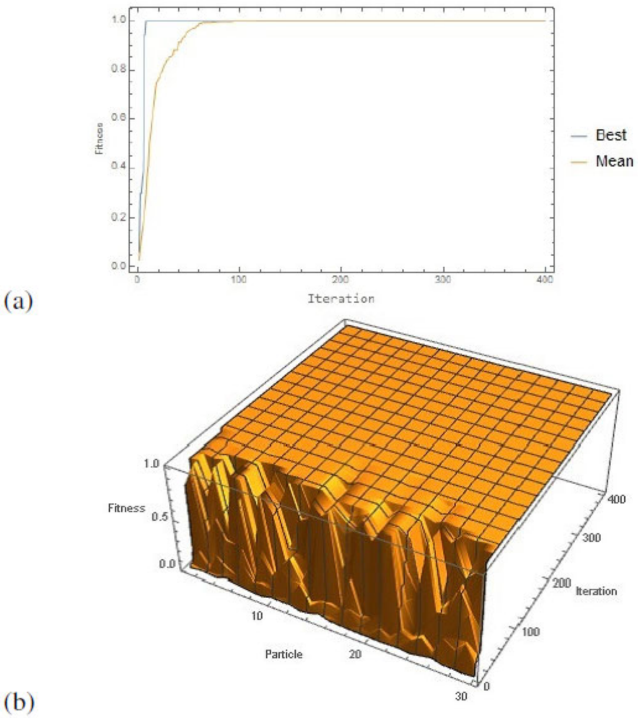
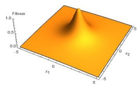
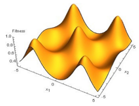
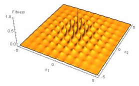
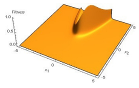
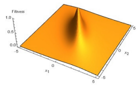


Figure 2. Search result of performing the PSO for Sphere problem in 5D caseD (a) The change of the best and average value in the search process, (b) The change of fitness value of each particle

Table 1. Functions and criteria of the given suite of benchmark problems. The search space for each benchmark problem is limited to $X \in (-5.12, 5.12)^D$.

Problem	Function	Criterion	Distribution in 2D
Sphere (*Sp*)	$f_{Sp}(\vec{x}) = \sum_{d=1}^{D} x_d^2$	$g_{Sp}(\vec{x}) = \frac{1}{1+f_{Sp}(\vec{x})}$	
Griewank (*Gr*)	$f_{Gr}(\vec{x}) = \frac{1}{4000}\sum_{d=1}^{D} x_d^2 - \prod_{d=1}^{D} cos\left(\frac{x_d}{\sqrt{d}}\right) + 1$	$g_{Gr}(\vec{x}) = \frac{1}{1+f_{Gr}(\vec{x})}$	
Rastrigin (*Ra*)	$f_{Ra}(\vec{x}) = \sum_{d=1}^{D}\left(x_d^2 - 10\, cos\,(2\pi x_d) + 10\right)$	$g_{Ra}(\vec{x}) = \frac{1}{1+f_{Ra}(\vec{x})}$	
Rosenbrock (*Ro*)	$f_{Ro}(\vec{x}) = \sum_{d=1}^{D-1}\left[100\left(x_{d+1} - x_d^2\right)^2 + \left(1 - x_d\right)^2\right]$	$g_{Ro}(\vec{x}) = \frac{1}{1+f_{Ro}(\vec{x})}$	
Schwefel (*Sc*)	$f_{Sc}(\vec{x}) = \sum_{d=1}^{N}\left(\sum_{j=1}^{d} x_j\right)^2$	$g_{Sc}(\vec{x}) = \frac{1}{1+f_{Sc}(\vec{x})}$	

Comparing the obtained search results of each search method for dealing with $Sphere$ problem in 5D case, it is found that the PSO and PSOIW relatively have better search performance among the used search methods of PSO. In addition, it is found that DE/best/1 has better search performance among the used search methods of DE.

Comparing the obtained search results of each search method for dealing with $Griewank$ problem in 5D case, it is found that DE/best/1 has better search performance among the used search methods of DE. In addition, we can see that PSOIW relatively has better search performance among the used search methods of PSO.

Comparing the obtained search results of each search method for dealing with $Rastrigin$ problem in 5D case, it is found that the PSO has better search performance among the used search methods of PSO. And it is found that DE/rand/2 relatively has better search performance among the used search methods of DE.

Comparing the obtained search results of each search method for dealing with $Rosenbrock$ problem in 5D case, it is found that DE/best/2 has better search performance among the used search methods of DE. In addition, it is found that PSOIW relatively has better search performance among the used search methods of PSO.

Comparing the obtained search results of each search method for dealing with $Schwefel$ problem in 5D case, it is found that the PSO and PSOIW relatively have better search performance among the used search methods of PSO. In addition, it is found that DE/best/1 has better search performance among the used search methods of DE.

Table 2. The major parameters used in implementing the search methods to explore the best solution

Parameter	Value
Number of individuals or particles, N	30
Number of dimension, D	5, 10
Rate of crossover operator, C_r	1.0
Number of search, K	400
Two points of crossover	—
Field of search, X	$(-5.12, 5.12)^D$

3.2 Comparison (2) of search performance in 10D case

For the comparison of search performance, we further increase the difficulty level of the given optimization problems to deal with the benchmark problems shown in Table 1 in 10D case. As the implementation in Section 3.1, the change in the search characteristics of search methods, i.e., DE/rand/1 and the PSO, are investigated.

Firstly, DE/rand/1 is implemented for dealing with $Sphere$ problem in 10D case. As an example, Figure 3 shows the obtained search results.

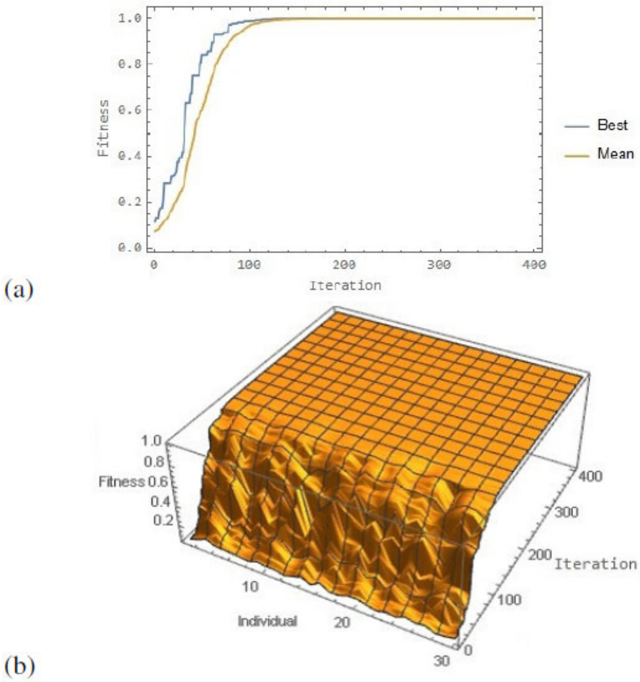


Figure 3. Search result of performing DE/rand/1 for Sphere problem in 10D case. (a) The change of the best and average value in the search process; (b) The change of fitness value of each individual.

We can see clearly that the search of DE/rand/1 is carried out smoothly, and the best solution of the given $Sphere$ problem is found out by the change state of the evaluative values of the best solution and average solution shown in Figure 3(a). And comparison with the search result of Figure 1(a), it is slowly search to the best solution with the increasing difficulty. Figure 3(b) indicates that the change of the evaluative value of each individual during the whole search process. Moreover, it can be confirmed from the search result of Figure 3(b) that the evaluative values (fitness) of each individual changes smoothly in the initial stage of the search for performing DE/rand/1.

Secondly, the PSO is implemented for dealing with $Sphere$ problem in 10D case. Figure 4 shows the obtained search results.

Comparing the obtained search results of both Figure 3(a) and Figure 4(a), it is clear that the PSO can find out the best solution quickly than DE/rand/1 does. The changes of the evaluative values (fitness) of the best solution is found out hastily in Figure 4(a). We can see that the search of the PSO is carried out to successfully found out find the best solution in 10D case. However, the comparison with the search situation in 5D case, it is obvious that the best solution is discovered slowly with increasing the number of dimension. And by comparing the change state of fitness values in Figure 3(b) and Figure 4(b), it is clear that the evaluative value (fitness) of each individual and each particle are smoothly changed.

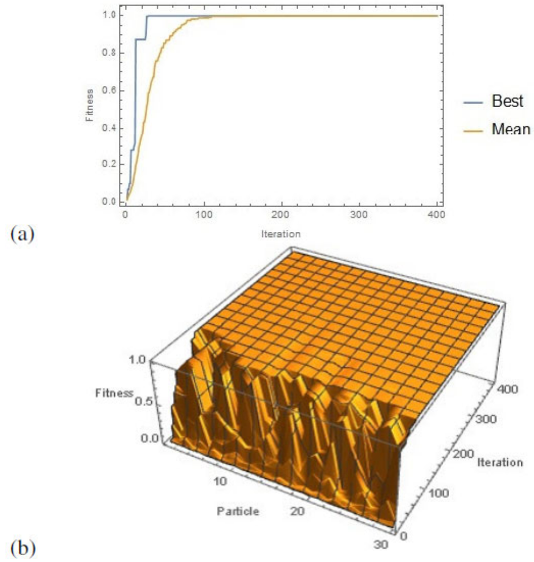


Figure 4. Search result of performing the PSO for Sphere problem in 10D case. (a) The change of the best and average value in the search process; (b) The change of fitness value of each particle.

By using each search method of DE and PSO, we carried out many computer experiments to handle the given benchmark problems in 10D case. As the obtained search results, the stochastic data of the obtained search results are listed in Table 4 for each benchmark problem.

Comparing the obtained search results of each search method for dealing with $Sphere$ problem in 10D case shown in Table 4, it is found that the PSO and PSOIW have the best search performance. We can also see that DE/best/1 relatively has better search performance among the used search methods of DE.

Comparing the obtained search results of each method for dealing with $Griewank$ problem in 10D case, it can be confirmed that DE/best/1 has better search performance among the used search methods of DE. And it also can be confirmed that the PSO has better search performance among the used search methods of PSO.

Comparing the obtained search results of each method for dealing with $Rastrigin$ problem in 10D case, it can be confirmed that DE/rand/1 has better search performance among the used search methods of DE. And it can be confirmed that the PSO has better search performance among the used search methods of PSO.

Comparing the obtained search results of each method for dealing with $Rosenbrock$ problem in 10D case, we can see that DE/best/1 has better search performance than the used other search methods of DE. And the PSO has better search performance among the used search methods of PSO. In this case, DE/best/1 has better search performance than that the PSO has.

Comparing the obtained search results of each method for dealing with $Schwefel$ problem in 10D case, it is found that the PSO and PSOIW have the best search performance. We can also see that DE/best/1 relatively has better search performance among the used search methods of DE.

Based on the obtained discrimination results of Table 3 and Table 4, Figure 5 shows the selected number of judgments of each search method for dealing with each benchmark problem. Therefore, the high search performance of the PSO and DE/best/1 can be confirmed.

Table 3. Search results for dealing with each benchmark problem in 5D case

Problem	Method	Max.	Mean	S.D.	Judgment
Sp	DE/rand/1	0.9999	0.9999	$1.6 * 10^{-15}$	
	DE/rand/2	0.9999	0.9999	$1.1 * 10^{-15}$	
	DE/best/1	0.9999	0.9999	$3.8 * 10^{-16}$	$\surd$
	DE/best/2	0.9999	0.9999	$6.5 * 10^{-16}$	
	The PSO	1.0	1.0	0.0	$\surd$
	PSOIW	1.0	1.0	0.0	$\surd$
	CPSO	0.9999	0.9999	$2.6 * 10^{-15}$	
Gr	DE/rand/1	0.9999	0.9999	$2.8 * 10^{-15}$	
	DE/rand/2	0.9999	0.9999	$2.8 * 10^{-9}$	
	DE/best/1	1.0	0.9999	$5.5 * 10^{-16}$	$\surd$
	DE/best/2	0.9999	0.9972	0.0088	
	The PSO	1.0	0.8753	0.1310	
	PSOIW	1.0	0.9247	0.1060	$\surd$
	CPSO	0.9999	0.9117	0.3453	
Ra	DE/rand/1	1.0	0.9501	0.1577	
	DE/rand/2	0.9999	0.9999	$1.2 * 10^{-11}$	$\surd$
	DE/best/1	1.0	0.5843	0.2251	
	DE/best/2	1.0	0.8503	0.2409	
	The PSO	1.0	1.0	0.0	$\surd$
	PSOIW	1.0	0.6794	0.4172	
	CPSO	0.3344	0.1525	0.0824	
Ro	DE/rand/1	0.9934	0.9506	0.0367	
	DE/rand/2	0.9336	0.8464	0.0829	
	DE/best/1	0.9999	0.7608	0.3850	
	DE/best/2	0.9999	0.9997	0.0006	$\surd$
	The PSO	0.9999	0.7041	0.3336	
	PSOIW	0.9999	0.9923	0.0244	$\surd$
	CPSO	0.9999	0.6731	0.4148	
Sc	DE/rand/1	1.0	0.9999	$4.5 * 10^{-7}$	
	DE/rand/2	0.9999	0.9999	0.0	
	DE/best/1	1.0	1.0	$3.0 * 10^{-16}$	$\surd$
	DE/best/2	1.0	1.0	$1.1 * 10^{-15}$	
	The PSO	1.0	1.0	0.0	$\surd$
	PSOIW	1.0	1.0	0.0	$\surd$
	CPSO	1.0	1.0	$3.9 * 10^{-15}$	

4. Analysis of search results

According to the experimental results which deal with the five benchmark problems, we can see clearly that the obtained search performance and characteristics of each search method of both DE and PSO technologies. Through many computer experiments which handled the given five benchmark problems in 5D and 10D cases, respectively, the following result analysis can be acquired.

- Regarding search performance, DE/best/1 and the PSO relatively have better search performance regardless of each difficulty level of each given benchmark problem compared to the used other search methods. And the search capability of the PSO is faster than that of DE/best/1.
- In the search characteristics, DE/rand/1 is a fluctuation in the evaluative value of each individual in the initial stage of whole search process, and in construct, the PSO is fluctuation in the evaluative value of each particle in the initial stage of whole search process.
- Regarding search performance, the obtained search results of DE/rand/2 and DE/best/2 do not grow up so much. Therefore, there is a limit to the search performance in the introduction of a more complicated mechanism for DE search methods.
- PSOIW and CPSO that emphasize search accuracy and convergence tend to relatively fall into local solution in order to handle the given complex benchmark problem.

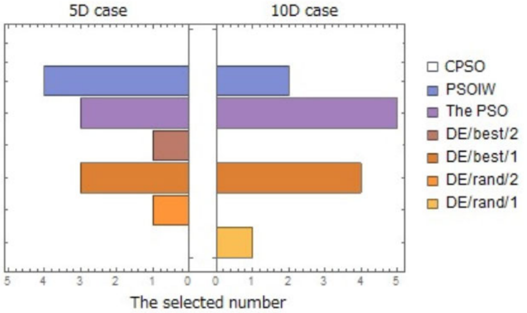


Figure 5. The selected number of judgment for dealing with each benchmark problem

Whether DE/best/1 or the PSO, the information of the best solution is used in the whole search process for reinforcing search capability to deal with difficult problems is a key hint. Since the information on the best solution is used in both of the PSO and DE/best/1, it is clear that the elitism strategy in the field of evolutionary computation plays an important role in their stochastic population search approaches.

Table 4. Search results for dealing with each benchmark problem in 10D case

Problem	Method	Max.	Mean	S.D.	Judgment
Sp	DE/rand/1	0.9999	0.9999	$7.0 * 10^{-11}$	
	DE/rand/2	0.9999	0.9999	$5.2 * 10^{-8}$	
	DE/best/1	0.9999	0.9999	$1.5 * 10^{-15}$	$\surd$
	DE/best/2	0.9999	0.9999	$6.3 * 10^{-15}$	
	The PSO	1.0	1.0	0.0	$\surd$
	PSOIW	1.0	1.0	0.0	$\surd$
	CPSO	0.9997	0.9790	0.0311	
Gr	DE/rand/1	0.9999	0.9999	$9.1 * 10^{-11}$	
	DE/rand/2	0.9999	0.9999	0.0000	
	DE/best/1	0.9999	0.9999	$1.5 * 10^{-15}$	$\surd$
	DE/best/2	0.9999	0.9944	0.0117	
	The PSO	1.0	0.9315	0.1294	$\surd$
	PSOIW	1.0	0.9157	0.1447	
	CPSO	0.9995	0.9029	0.1125	
Ra	DE/rand/1	0.9999	0.9499	0.1576	$\surd$
	DE/rand/2	0.8157	0.4186	0.2110	
	DE/best/1	0.3344	0.1809	0.0710	
	DE/best/2	0.9999	0.4301	0.2404	
	The PSO	1.0	0.6163	0.4953	$\surd$
	PSOIW	0.1830	0.0649	0.0528	
	CPSO	0.1668	0.0609	0.0447	
Ro	DE/rand/1	0.2617	0.2184	0.0272	
	DE/rand/2	0.1908	0.1762	0.0092	
	DE/best/1	0.9932	0.8856	0.0852	$\surd$
	DE/best/2	0.9662	0.5170	0.1658	
	The PSO	0.5766	0.1397	0.1568	$\surd$
	PSOIW	0.1431	0.1139	0.0118	
	CPSO	0.1449	0.0786	0.0431	
Sc	DE/rand/1	0.9532	0.9020	0.0439	
	DE/rand/2	0.9039	0.8024	0.0696	
	DE/best/1	1.0	1.0	$4.0 * 10^{-10}$	$\surd$
	DE/best/2	0.9999	0.9988	0.0009	
	The PSO	1.0	1.0	0.0	$\surd$
	PSOIW	1.0	1.0	0.0	$\surd$
	CPSO	0.9875	0.8536	0.1704	

5. Handling the 2-bit and 3-bit parity problems

Based on the above obtained experimental results, we try to implement the PSO as a deputy search method of both DE and PSO technologies to handle the 2-bit and 3-bit parity problems, to confirm the search performance of the findings and know-how, and observe the relational results by building a multilayer neural network.

Figure 6 shows a simple three-layer neural network. This one consists of an input layer, a hidden layer, and an output layer, the interconnected by the modified weights and biases, represented by links between layers. Furthermore, there is a single bias unit that is connected to each unit other than the input units.

Table 5 gives the input/target patterns of the 2-bit parity (i.e., XOR) problem. $\vec{x} = \{x_1, x_2\}$ is an input vector, and $t(\vec{x})$ is a teacher target, i.e., it gives the input vector which belongs to 0-class or 1-class.

For dealing with the given the parity problems, we have to use the non-linear feed-forward neural network to express the discriminant function as follows:

$$o(\vec{x}|\vec{w}) = f\left(\sum_{j=1}^{Hn} w_{kj} f\Big(\sum_{i=1}^{D} w_{ji}x_i + w_{j0}\Big) + w_{k0}\right) \quad (14)$$

where $f(\cdot)$ is a logistic function, i.e., $f(z) = \frac{1}{1+e^{-3z}}$. Hn is the number of the hidden units, w_{kj}, w_{ji}, w_{k0}, w_{j0} are the interconnected weights between hidden layer to output layer, the interconnected weights between input layer to hidden layer, bias of output unit, and biases of hidden units, respectively.

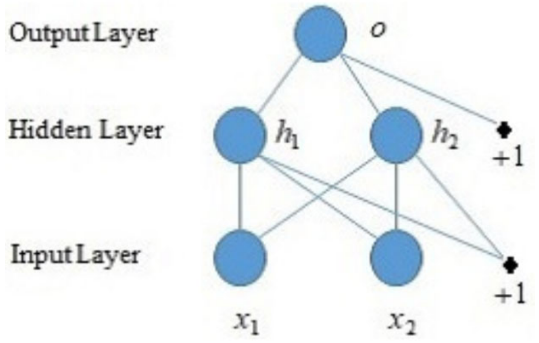


Figure 6. A construction of a three-layer neural network for handling the 2-bit parity problem

Table 5. 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

The error between the teacher target and output of the neural network for the given 2-bit parity problem is expressed as follows:

$$e(\vec{x}) = \frac{1}{2N}\sum_{i=1}^{N}\big(t_i(\vec{x}) - o_i(\vec{x}|\vec{w})\big)^2 \quad (15)$$

where $e(\vec{x})$ is called as to mean squared error (MSE), N is the number of the given input/target patterns. In this case, the evaluative criterion is given as follows:

$$g(\vec{x}) = \frac{1}{1+e(\vec{x})} \quad (16)$$

Obviously, when the value of MSE becomes zero, the given parity problem can be solved with high accuracy.

Since the two units in hidden layer are used, total of the interconnected weights and biases is nine (i.e., this is a search problem in 9D). Then the nine interconnected weights and biases should be arranged to a vector in order for implementing the PSO to the pattern classification.

5.1 Handling the 2-bit parity problem (1)

In this case, the number of the hidden layer is set to 2 units to investigate their role in the pattern classification.

As an example of the experimental data, 10 iterative searches were performed for the computational experiment to deal with the 2-bit parity problem, the obtained search result is shown in Figure 7.

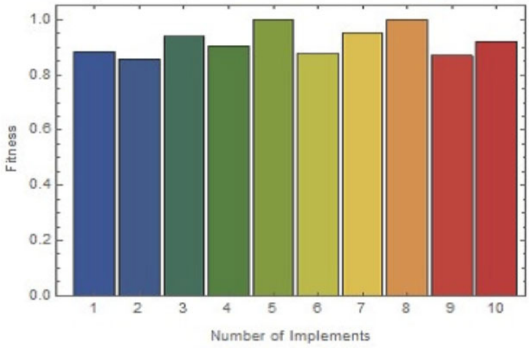


Figure 7. The fitness value for each number of searching

The distribution of the evaluative values for each number of searches is shown in Figure 7. In this case, the maximum value of the fitness is 1.0, the average value is 0.9212, and the standard deviation is 0.0516 for the search results of performing 10 experiments.

As an example of the experimental data, Figure 8 shows the obtained search result which has the fitness value is 1.0 for dealing with the 2-bit parity problem.

According to the situation change of the fitness value of the best solution in the search process in Figure 8(a), it is clear that the classification by implementing the PSO is success, two class points are fully classified with high accuracy. And the distinguish speed is very fast to find out the best solution for the given pattern classification.

Figure 8(b) gives the vector of the final interconnected weights and biases of the neural network, which well achieve the goal of handling the 2-bit parity problem.

Figure 8(c) shows that the distinguish relation of the two hidden units. We can see clearly that one hidden unit deals with the 2-bit AND problem, and the other unit deals with the 2-bit OR problem.

Figure 8(d) finally gives the distinguish surface of classification, which completely deals with the 2-bit parity (XOR) problem.

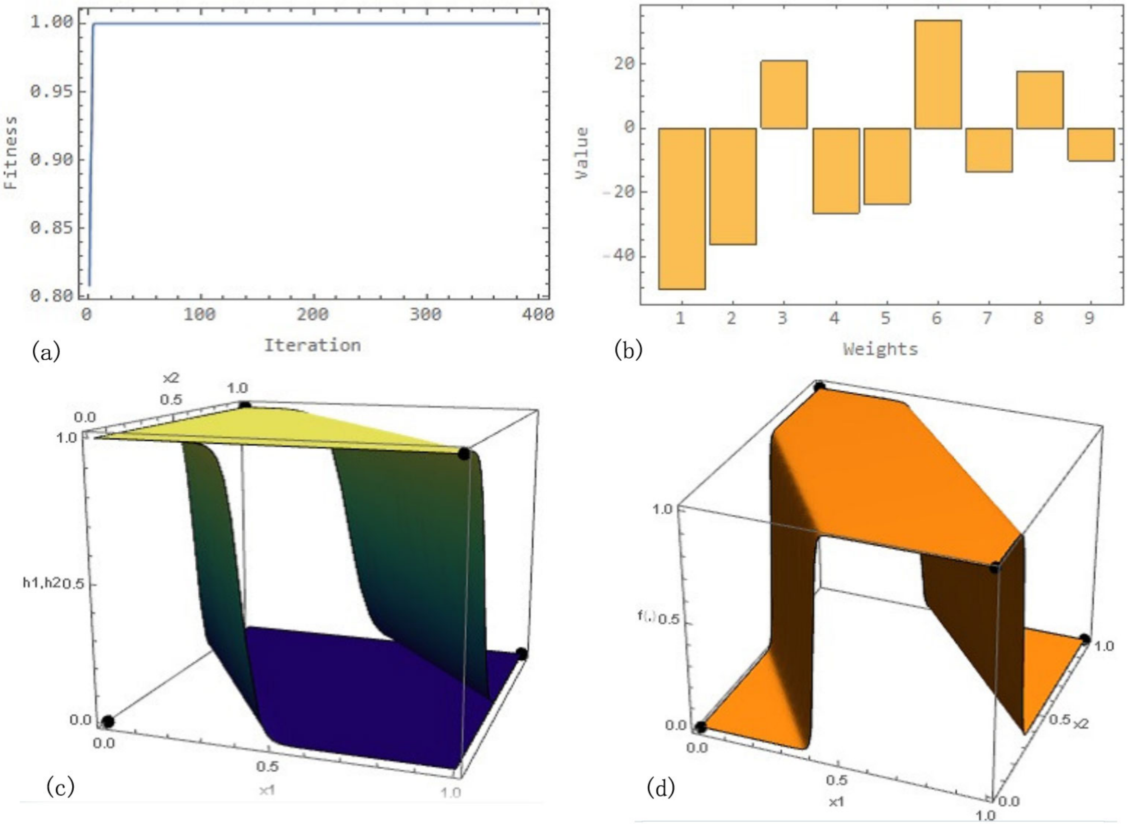


Figure 8. An experimental search of implementing the three-layer neural network for handling the 2-bit parity problem. (a) the search process of the fitness of the best solution; (b) the values of the final interconnected weights and biases of the neural network; (c) the distinguish surface of classification in hidden layer; (d) the distinguish surface of classification in output layer.

5.2 Handling the 2-bit parity problem (2)

In this case, the number of the hidden layer is set to 3 units to investigate their role for the pattern classification. Since the three units in hidden layer are used, total of the interconnected weights and biases is thirteen (i.e., this is a search problem in 13D).

As an example of the experimental data, Figure 9 shows the obtained search results for dealing with the 2-bit parity problem. Generally, the search difficulty becomes big as the search space increases..

However, we can see that Figure 9(a) shows that the distinguish speed is very fast than that shown in Figure 8(a) to find out the best structure of the neural network according to the situation change of the fitness value of the best solution. And the structure of the neural network becomes complex one, but it can be seen that the PSO search found out the best solution with the least iteration.

Generally, the difficulty of the search increases as the search space expands. But, this concept does not hold for the search results of the PSO in this case.

Figure 9(b) gives the vector of the final interconnected weights and biases of the neural network, which arrives at the target, i.e., solving the 2-bit XOR problem. The redundant hidden unit just plays a redundant role which deals with the 2-bit AND problem or 2-bit OR problem.

Figure 9(c) indicates that distinguish relation of the three hidden units. Similar to the above mentioned in Section 5.1, they deal with the 2-bit AND problem, and the 2-bit OR problem, respectively.

Finally, Figure 9(d) shows that the distinguish surface of the pattern classification, which completely deals with the 2-bit XOR problem with high accuracy.

5.3 Handling the 3-bit parity problem

Considering that deal with the 3-bit parity problem by implementing the PSO. Table 6 gives the input/target patterns of the 3-bit parity problem.

In this case, the number of the hidden layer is set to 3 units to investigate their role in the pattern classification. Since the three units in hidden layer are used, total of the interconnected weights and biases is sixteen (i.e., this is a search problem in 16D).

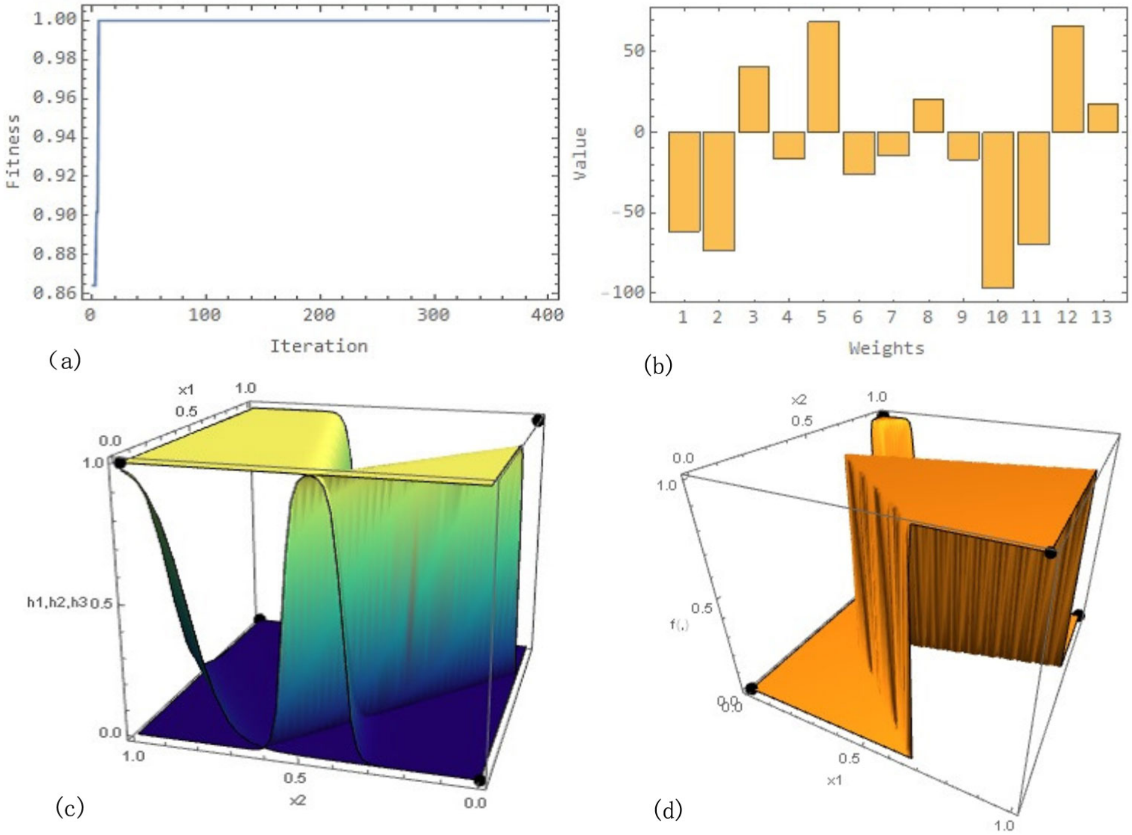


Figure 9. An experimental search of implementing the three-layer neural network for handling the 2-bit parity problem. (a) the search process of the fitness of the best solution; (b) the values of the final interconnected weights and biases of the neural network; (c) the distinguish surface of classification in hidden layer; (d) the distinguish surface of classification in output layer.

As an example of the experimental data, Figure 10 shows the obtained search results for dealing with the 3-bit parity problem. Based on the search situation of the best solution shown in Figure 10(a), we can confirm that the 3-bit parity problem is successfully classified. And Figure 10(b) gives the vector of the final interconnected weights and biases of the obtained neural network, which arrives at the target, i.e., successfully solving the 3-bit parity problem with high accuracy.

Table 6. 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, the 3-bit parity problem becomes a hard one for exploring, the success rate of the PSO's search is very low ($< 3\%$) in the complete discrimination case.

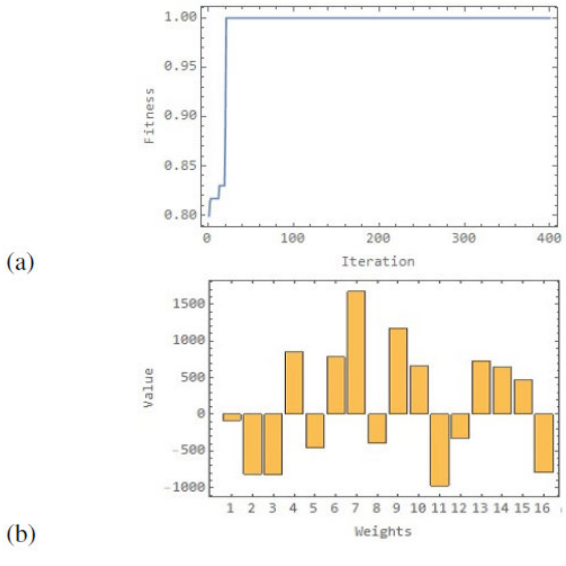


Figure 10. An experimental search of implementing the three-layer neural network for handling the 3-bit parity problem. (a) the search process of the fitness of the best solution; (b) the values of the final interconnected weights and biases of the neural network.

6. CONCLUSION

In this paper, we systematically compared and investigated the search performance and characteristics of both DE and PSO technologies. Specially, through massive computer experiments to deal with the given five benchmark problems in 5D and 10D cases, the obtained search results of the search performance and characteristics of DE and PSO, i.e., DE/rand/1, DE/rand/2, DE/best/1, DE/best/2, the PSO, PSOIW, and CPSO were analyzed in detail and quantitative comparison were implemented, respectively.

According to the obtained search performance of two search methods, i.e., DE/best/1 and the PSO, are found out to be relatively better search performance among the used search methods. In addition, the PSO has a character that the search speed is fast with contrasting to DE/best/1. Since the information on the best solution was used in both the PSO and DE/best/1, it is clear that the elitism strategy in the field of evolutionary computation plays an important role in their stochastic population search approaches.

As the applicative examples, the PSO is used to handle typical 2-bit and 3-bit parity problems for the pattern classification. And the obtained experimental results showed that the PSO has high search capability.

Concretely, based on the findings and know-how of the obtained search results from our computer experiments, in order to further obtain the better search performance, they can provide an important reference and hint about how to select some powerful search methods for building rational and efficient hybrid multi-swarm search methods.[28,29] Specially, we will plan to implement a new type method of stochastic multi-population (multi-swarm) search to deal with the complicated optimization problems and applicative real-world problems.

REFERENCES

[1] Storn R, Price K. Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization, 1997; (11): 341-359. https://doi.org/10.1023/A:1008202821328

[2] Eberhart RC, Kennedy J. A new optimizer using particle swarm theory. Proceedings of the Sixth International Symposium on Micro Machine and Human Science. 1995; 39-43. Nagoya, Japan.

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

[4] Eltaeib T, Mahmood A. Differential Evolution: A Survey and Analysis. Appl. Sci.. 2018; 8(10) (1945). https://doi.org/10.3390/app8101945

[5] Valle Y, Venayagamoorthy GK, Mohagheghi S, Hernandez JC, Harley RG. Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems. IEEE Transactions on Evolutionary Computation. 2008; 12(2): 171-195.

[6] Wang BC, Li HX, Li JP, Wang Y. Composite Differential Evolution for Constrained Evolutionary Optimization. IEEE Transactions on Systems, Man, and Cybemetics: Systems. 2019; 49(Issue 7): 1482-1495. https://doi.org/10.1109/TSMC.2018.2807785

[7] Plagianakos VP, Tasoulis DK, Vrahatis MN. A Review of Major Application Areas of Differential Evolution. in: Chakraborty (Ed.): Advances in Differential Evolution. 2008; SCI 143: 197-238. https://doi.org/10.1007/978-3-540-68830-3_8

[8] Qin AK, Huang VL, Suganthan PN. Differential evolution algorithm with strategy for global numerical optimization. IEEE Transactions on Evolutionary Computation. 2009; 13(2): 398-417. https://doi.org/10.1109/TEVC.2008.927706

[9] Sar S, Roy A, Purkayasthy BS. Application of Particle Swarm Optimization in Data Clustering: A Survey. International Journal of Computer Applications. 2013; 65(25): 69-73.

[10] Goldberg DE. Genetic Algorithm in Search Optimization and Machine Learning, Reading. MA: Addison-Wesley; 1989.

[11] Beyer HG. The Theory of Evolution Stratagies. Springer; 2001.

[12] Kachitvichyanukul V. Comparison of Three Evolutionary Algorithms: GA, PSO, and DE. Industrial Engineering & Management Systems. 2012; 11(3): 215-223. https://doi.org/10.7232/iems.2012.11.3.215

[13] Eberhart RC, Shi Y. Comparing inertia weights and constriction factors in particle swarm optimization. Proceedings of the 2000 IEEE Congress on Evolutionary Computation. 2000; (1): 84-88. San Diego, CA, USA.

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

[15] Shi Y, Eberhart RC. Empirical study of particle swarm optimization. Proceedings of the IEEE Congress Evolutionary Computation. 1999; 3: 1945-1950.

[16] Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation. 2002; 6(1): 58-73. https://doi.org/10.1109/4235.985692

[17] Clerc M. Particle Swarm Optimization. Hermes Science Pubns, UK; 2006.

[18] 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.

[19] Sho H. Investigation of Particle Multi-Swarm Optimization with Diversive Curiosity. Engineering Letters. 2020; 28(3): 960-969. http://www.engineeringletters.com/issues v28/issue 3/index.html

[20] Ripley BD. Pattern Recognition and Neural Networks. 1996; Cambridge University Press, Great Britain.

[21] Hillier FS, Lieberman GJ. Introduction to Operations Research. Holden-Day, Inc., Oakland, CA, USA; 1986.

[22] Haykin S. Neural Networks: A Comprehensive Foundation. Prentice Hall, New York; 1998.

[23] Duda OR, Hart PE, Stork DG, Pattern Classification. John Wiley & Sons, Inc., USA; 2001.

[24] YHohil ME, Liu D, Smith SH. Solving the N-bit parity problem using neural netwoks. Neural Networks. 1999; 12(Issue 9): 1321-1323. https://doi.org/10.1016/S0893-6080(99)00069-6

[25] Ju´areg-Castillo E, P´erez-Castro N, Mezura-Montes E. An Improved Centroid-Based Boundary Constraint-Handling Method in Differential Evolution for Constrained Optimization. International Journal of Pattern Recognition and Artificial Intelligence. 2017; 1-38. doi:10.1142/S0218001417590236.

[26] Moscato P. On Evolution, Search Optimization, Genetic Algorithmsand Martial Arts: Towards Memetic Algorithms. Technical Report Caltech Concurrent Computation Program, Report 826. 1989; California Institute of Technology, Pasadena, CA 91125, USA.

[27] Suganthan PN, Hansen N, Liang JJ, Deb K, Chen YP, Auger A, Tiwari S. Problem Definitions and Evaluation Criteria for the CEC 2005. http//:www.ntu.edu.sg/home/epnsugan/index files/CEC-05/Tech-Report-May-30-05.pdf

[28] 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.

[29] Sho H. Investigation of Particle Multi-Swarm Optimization with Diversive Curiosity. Engineering Letters. 2020; 28(3): 960-969. http://www.engineeringletters.com/issues v28/issue 3/index.html