View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by CiteSeerX

Selecting Strategies in Particle Swarm Optimization by Sampling-Based Landscape Modality Detection

Tetsuyuki Takahama[1] **and Setsuko Sakai**[2]

[1] Department of Intelligent Sciences, Hiroshima City University, Hiroshima, Japan

[2] Faculty of Commercial Sciences, Hiroshima Shudo University, Hiroshima, Japan

Abstract— *If the landscape of the objective function is unimodal, the efficiency of population-based optimization algorithms (POAs) can be improved by selecting strategies for local search around a best solution. If the landscape is multimodal, the robustness of the POAs can be improved by selecting strategies for global search in search space. We have proposed a method that estimates the landscape modality by sampling the objective values along a line and counting the number of changes in the objective values from increasing to decreasing and vice versa. In this study, in order to improve the performance of particle swarm optimization (PSO), we propose to select a proper strategy according to the landscape modality: The gbest model is selected in unimodal landscape and the lbest model is selected in multimodal landscape. The advantage of the proposed method is shown by solving unimodal and multimodal problems and by comparing it with standard PSOs.*

Keywords: Particle swarm optimization, Landscape modality, Landscape modality estimation, Lbest model, Gbest model

1. Introduction

There exist many studies on solving optimization problems using population-based optimization algorithms (POAs) in which a population or multiple search points are used to search for an optimal solution. For example, swarm intelligence algorithms inspired by collective animal behavior have been studied such as particle swarm optimization (PSO)[1], [2] and ant colony optimization. Also, evolutionary algorithms inspired by biological evolution have been studied such as genetic algorithm, evolution strategy and differential evolution[3], [4]. In general, POAs are stochastic direct search methods, which only need function values to be optimized, and are easy to implement. For this reason, POAs have been successfully applied to various optimization problems.

In this study, we paid attention to improve PSO. There are two models or movement strategies in PSO: the *gbest* model where each search point or a particle moves toward the best point in the population and the *lbest* model where each search point moves toward a best point in the neighbor points. It is known that the gbest model can solve unimodal problems efficiently but the strategy cannot solve multimodal problems stably and the search by the strategy is sometimes trapped at a local optimal solution. In contrast, it is known that the lbest strategy is robust to multimodal problems but the strategy cannot solve unimodal problems efficiently. However, the landscape of a problem to be optimized is often unknown and the landscape is changing dynamically while the search process proceeds. Thus, it is difficult to select a proper strategy.

We have proposed a simple method that detects the modality of landscape being searched: unimodal or not unimodal[5], [6], [7]. In the method, some points on the line connecting between the centroid of search points and the best search point are sampled. When the objective values of the sampled points are changed decreasingly and then increasingly, it is thought that one valley exists. If there exists only one valley or the landscape is unimodal, the gbest strategy is adopted. In this case, it is expected that the strategy can realize efficient search. If the number of valley is greater than one, the lbest strategy is adopted. In this case, it is expected that the strategy improves the divergence of the search and prevents premature convergence. The effect of the proposed method is shown by solving 13 benchmark problems including unimodal problems and multimodal problems.

In Section 2, related works are briefly reviewed. Detecting landscape modality is explained in Section 3. The optimization problem is defined and PSO is explained in Section 4. PSO with detecting landscape modality is proposed in Section 5. In Section 6, experimental results on some problems are shown. Finally, conclusions are described in Section 7.

2. Related Works

Many studies on strategy selection and parameter tuning have been done in order to improve the efficiency. The studies can be classified into two main categories: observation-based and success-based control[5], [6], [7].

1) observation-based control: The current search state is observed, proper strategies or parameter values are inferred according to the observation, and strategies and/or parameters are dynamically controlled. FADE(Fuzzy Adaptive DE)[8] observes the movement of search points and the change of function values between successive generations, and controls algorithm parameters. DESFC(DE with Speciation and Fuzzy

Clustering)[9] adopts fuzzy clustering, observes partition entropy of search points, and controls a parameter and the mutation strategies between the rand and the species-best strategy.

2) success-based control: It is recognized as a success case when a better search point than the parent is generated. The strategies and/or parameters are adjusted so that the values in the success cases are frequently used. It is thought that the self-adaptation, where strategies and/or parameters are contained in individuals and are evolved by applying evolutionary operators to the parameters, is included in this category. DESAP(Differential Evolution with Self-Adapting Populations)[10] controls algorithm parameters including population size self-adaptively. SaDE(Self-adaptive DE)[11] controls the selection probability of the mutation strategies according to the success rates and controls the mean value of a crossover rate for each strategy according to the mean value in success case. JADE(adaptive DE with optional external archive)[12] and MDE_pBX(modified DE with p-best crossover)[13] control the mean and power mean values of two parameters according to the mean values in success cases.

In the category 1), it is difficult to select proper type of observation which is independent of the optimization problem and its scale. In the category 2), when a new good search point is found near the parent, parameters are adjusted to the direction of convergence. In problems with ridge landscape or multimodal landscape, where good search points exist in small region, parameters are tuned for small success and big success will be missed. Thus, search process would be trapped at a local optimal solution.

In this study, we propose a new observation-based control in the category 1). As a problem independent observation, landscape modality is adopted and it is estimated whether the problem is unimodal or multimodal using sampling. It is thought that a proper strategy or algorithm parameters can be selected if the landscape modality can be identified.

3. Detecting Landscape Modality using Sampling

Search points in a current population or a set of search points $P = \{\boldsymbol{x}_i | i = 1, 2, \cdots, N\}$ are used to detect landscape modality using sampling[5], [6], where N is the number of search points or population size. The range of search points is determined, a line is drawn in the range, and equally spaced points are sampled along the line.

3.1 Sampling

The objective values are examined along the following line, which connects the centroid of search points $\boldsymbol{x}^g$ and the best search point $\boldsymbol{x}^b$.

$$\boldsymbol{x} = \boldsymbol{x}^g + \lambda(\boldsymbol{x}^b - \boldsymbol{x}^g) \tag{1}$$

$$\boldsymbol{x}^g = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i \tag{2}$$

$$\boldsymbol{x}^b = \arg\min_{\boldsymbol{x}_i \in P} f(\boldsymbol{x}_i) \tag{3}$$

where λ is a parameter for deciding the position of a point on the line. The range of the search points $[\boldsymbol{x}^{\min}, \boldsymbol{x}^{\max}]$ can be given as follows:

$$x_j^{\min} = \min_i x_{ij} \tag{4}$$

$$x_j^{\max} = \max_i x_{ij} \tag{5}$$

The range of the λ, $[\lambda^{\min}, \lambda^{\max}]$ satisfies the following condition:

$$x_j^{\min} \le x_j^g + \lambda(x_j^b - x_j^g) \le x_j^{\max} \tag{6}$$

Thus, if $(x_j^b - x_j^g)$ is positive, the range of the λ is given by:

$$\lambda^{\min} = \max_j \frac{x_j^{\min} - x_j^g}{x_j^b - x_j^g} \tag{7}$$

$$\lambda^{\max} = \min_j \frac{x_j^{\max} - x_j^g}{x_j^b - x_j^g} \tag{8}$$

If $(x_j^b - x_j^g)$ is negative, $x_j^{\min}$ and $x_j^{\max}$ in the equations are exchanged.

In order to decide M sampling points $\{\boldsymbol{x}_k | k = 1, 2, \cdots, M\}$, λ_k is given as follows:

$$\lambda_k = \lambda^{\min} + \frac{\lambda^{\max} - \lambda^{\min}}{M-1}(k-1) \tag{9}$$

$$\boldsymbol{z}_k = \boldsymbol{x}^g + \lambda_k(\boldsymbol{x}^b - \boldsymbol{x}^g) \tag{10}$$

Figure 1 shows an example of the sampling, where search points are shown by black circles, the centroid is shown by a white circle, sampling points are shown by triangles in case of $M = 6$.

3.2 Landscape Modality

In the obtained sequence $\{f(\boldsymbol{z}_k) | k = 1, 2, \cdots, M\}$, hill-valley relation is examined. For each point, the function $dir(\cdot)$ is introduced in order to judge whether the change is increasing or decreasing:

$$dir(\boldsymbol{z}_k) = \begin{cases} 1 & (f(\boldsymbol{z}_{k+1}) > f(\boldsymbol{z}_k)) \\ -1 & (f(\boldsymbol{z}_{k+1}) < f(\boldsymbol{z}_k)) \\ dir(\boldsymbol{z}_{k-1}) & (\text{otherwise}) \end{cases} \tag{11}$$

Figure 2 shows an example of detecting unimodal landscape, where the objective values are shown by the function of λ.

The landscape modality is identified using the number of changes in dir function. If the value of dir changed from -1 to 1 only once or there is no changes, it is thought that one valley exists and the landscape is unimodal. Otherwise, the landscape is not unimodal.

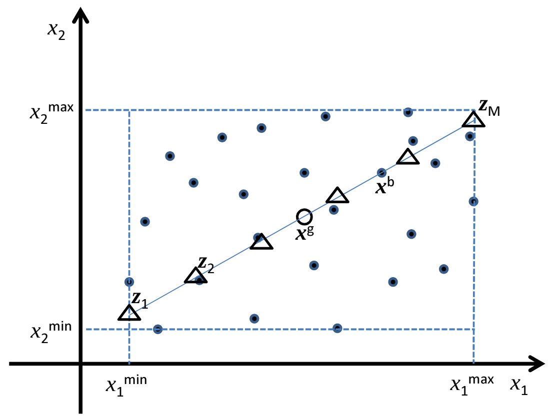

Fig. 1: An example of sampling for detecting landscape modality.

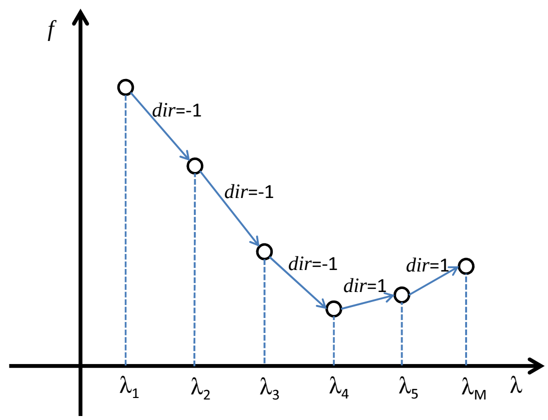

Fig. 2: An example of detecting unimodal landscape.

4. Optimization Problems and Particle Swarm Optimization

4.1 Optimization Problems

In this study, the following optimization problem (P) with lower bound and upper bound constraints will be discussed.

$$\begin{array}{lll}(\mathrm{P}) & \text{minimize} & f(\boldsymbol{x}) \\ & \text{subject to} & l_i \le x_i \le u_i,\ i=1,\dots,n,\end{array} \quad (12)$$

where $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)$ is an n dimensional vector and $f(\boldsymbol{x})$ is an objective function. The function f is a nonlinear real-valued function. Values l_i and u_i are the lower bound and the upper bound of x_i, respectively. Let the search space in which every point satisfies the lower and upper bound constraints be denoted by $\mathcal{S}$.

4.2 Particle Swarm Optimization

An animal such as an ant, a fish, and a bird has limited memory and ability to perform simple actions. In contrast, a group of animals such as an ant swarm, a fish school, and a bird flock can take complex or intelligent actions such as avoiding predators and seeking foods efficiently. Swarm intelligence is defined as the collective actions of agents that act autonomously and communicate each other. PSO[2] is a swarm intelligence based optimization method which was inspired by the movement of a bird flock. PSO imitates the movement to solve optimization problems and is considered as a population-based stochastic search method or POA.

Searching procedures by PSO can be described as follows: A group of agents minimizes the objective function f. At any time t, each agent i knows its current position $\boldsymbol{x}_i^t$ and velocity $\boldsymbol{v}_i^t$. It also remembers its personal best visited position until now $\boldsymbol{x}_i^*$ and the objective value $pbest_i$.

$$\boldsymbol{x}_i^* = \arg\min_{\tau=0,1,\cdots,t} f(\boldsymbol{x}_i^\tau) \quad (13)$$
$$pbest_i = f(\boldsymbol{x}_i^*) \quad (14)$$

Two models, gbest model and lbest model have been proposed. In the gbest model, every agent knows the best visited position $\boldsymbol{x}_G^*$ in all agents and its objective value $gbest$.

$$\boldsymbol{x}_G^* = \arg\min_i f(\boldsymbol{x}_i^*) \quad (15)$$
$$gbest = f(\boldsymbol{x}_G^*) \quad (16)$$

In the lbest model, each agent knows the best visited position $\boldsymbol{x}_l^*$ in the neighbors and its objective value $lbest_i$, where l is the best visited position in the neighborhood.

$$\boldsymbol{x}_l^* = \arg\min_{k\in N_i} f(\boldsymbol{x}_k^*) \quad (17)$$
$$lbest_i = f(\boldsymbol{x}_l^*) \quad (18)$$

where N_i is the set of neighbor agents to i. The velocity of the agent i at time $t+1$ is defined as follows:

$$v_{ij}^{t+1} = wv_{ij}^t + c_1\, rand_{1ij}\,(x_{ij}^* - x_{ij}^t) + c_2\, rand_{2ij}\,(x_{lj}^* - x_{ij}^t) \quad (19)$$

where $l=G$ in the gbest model, w is an inertia weight and $rand_{kij}$ is a uniform random number in $[0,1]$ which is generated in each dimension. c_1 is a cognitive parameter, c_2 is a social parameter which represent the weight of the movement to the personal best and the group/neighbors best respectively.

The position of the agent i at time $t+1$ is given as follows:

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \boldsymbol{v}_i^{t+1} \quad (20)$$

4.3 Algorithm of PSO

The algorithm of PSO is defined as follows:

1) Initializing agents: Each agent i with a position $\boldsymbol{x}_i$ and a velocity $\boldsymbol{v}_i$ is created. $\boldsymbol{x}_i$ is randomly generated in the search space S, namely each element x_{ij} is a uniform random number in $[l_j,u_j]$. $\boldsymbol{v}_i$ is the zero vector where every element v_{ij}=0 in this study. The best visited position is set to the initial position, namely $\boldsymbol{x}_i^*$=$\boldsymbol{x}_i$.

2) Selecting the best agent: The id of the best agent G is decided.
3) Stopping if termination condition is satisfied: The algorithm is stopped when the number of function evaluations reaches the maximum number of evaluations FE_{max}.
4) Updating agents: The position and velocity of each agent i are updated according to Eq.(19) and Eq.(20), respectively. The each element of the velocity is truncated in $[-V_{max_j}, V_{max_j}]$. If the objective value of the new position is better than the personal best value, the personal best visited position is replaced with the new position. If the objective value of the new position is better than the group best value, the group's best visited position is replaced with the new position.
5) Go back to 3.

5. Proposed Method

In this section, a method of selecting a movement strategy dynamically is proposed for PSO.

5.1 Strategy selection

In general, if divergence of agents is kept to realize a global search, it can be avoided to be trapped at a local solution but the efficiency of the search will be reduced. If convergence of agents is enforced to realize local search around the best agent, the efficiency of the search is improved but the search will be trapped at a local solution.

In PSO, the gbest model can realize the local search and the lbest model can realize the global search. In the lbest model, the neighborhood of agents is defined as a topology such as star topology, ring topology, mesh topology, and so on. In this study, the ring topology is adopted, where agents are connected in the order of the agent numbers. The neighborhood size $N_{neighbor}$ is an important parameter in the lbest model. Small neighborhood size strengthens the global search and large neighborhood size strengthens the local search. When the size is same as the population size, the lbest model becomes the gbest model. In this study, the gbest model is selected for unimodal landscape and the lbest model with $N_{neighbor} = 5$ including the agent itself is selected for multimodal landscape.

5.2 Proposed algorithm

Figure 3 shows the proposed algorithm named LPSO(PSO with detecting Landscape modality), where T_L is the interval of iterations when landscape modality is estimated, N_{small} is the neighborhood size for the global search, and N_{large} is the neighborhood size for the local search. Lines with '+' at the first column are the modification to standard PSO.

If the number of direction changes from decreasing to increasing and vice versa is 1 or zero, the landscape modality is estimated as unimodal. However, the estimation should be done carefully because the sampling is done in a small region and the number of sampling points is small. Thus, the number of successive unimodal estimations is counted and if the number is equal to or greater than $N_{unimodal}$ the landscape is identified as unimodal.

```
 Initialize P;
 Evaluate all x in P;
 G=arg min_{i|x_i∈P} f(x_i)
+unimodal=0;
 for(t=1;t ≤ T;t++) {
+  if(t%T_L==1) {
+    changed=landscape modality estimation in P;
+    if(changed==0 || changed==1) unimodal++;
+    else unimodal=0;
+  }
+  if(unimodal≥N_unimodal) N_neighbor=N_large;
+  else N_neighbor=N_small;
   for(each agent i in P) {
     l=best agent in i's neighborhood
            of size N_neighbor;
     for(each dimension j) {
       v_ij=wv_ij+c_1 rand_1ij (x*_ij-x_ij)
                +c_2 rand_2ij (x*_lj-x_ij);
       if(v_ij>V_max_j) v_ij=V_max_j;
       else if(v_ij<-V_max_j) v_ij=-V_max_j;
       x_ij=x_ij+v_ij;
     }
     Evaluate x_i;
     if(f(x_i) < f(x*_i)) {
       if(f(x_i) < f(x*_G)) G=i;
       x*_i=x_i;
     }
   }
 }
 returns x*_G as the best solution;
```

Fig. 3: Algorithm of LPSO.

6. Solving Optimization Problems

In this study, well-known thirteen benchmark problems are solved.

6.1 Test Problems and Experimental Conditions

The 13 scalable benchmark functions are shown in Table 1[12]. All functions have an optimal value 0. Some characteristics are briefly summarized as follows: Functions f_1 to f_4 are continuous unimodal functions. The function f_5 is Rosenbrock function which is unimodal for 2- and 3-dimensions but may have multiple minima in high dimension cases[14]. The function f_6 is a discontinuous step function, and f_7 is a noisy quartic function. Functions f_8 to f_{13} are multimodal functions and the number of their local minima increases exponentially with the problem dimension[15].

Independent 50 runs are performed for 13 problems. The dimension of problems is 30 (D=30). The maximum number of evaluations $FE_{\max}$ is 200,000. The parameters of PSO

Table 1: Test functions of dimension D. These are sphere, Schwefel 2.22, Schwefel 1.2, Schwefel 2.21, Rosenbrock, step, noisy quartic, Schwefel 2.26, Rastrigin, Ackley, Griewank, and two penalized functions, respectively[16].

Test functions	Domain
$f_1(\boldsymbol{x}) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]^D$
$f_2(\boldsymbol{x}) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	$[-10, 10]^D$
$f_3(\boldsymbol{x}) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j \right)^2$	$[-100, 100]^D$
$f_4(\boldsymbol{x}) = \max_i \{\lvert x_i \rvert\}$	$[-100, 100]^D$
$f_5(\boldsymbol{x}) = \sum_{i=1}^{D-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	$[-30, 30]^D$
$f_6(\boldsymbol{x}) = \sum_{i=1}^{D} \lfloor x_i + 0.5 \rfloor^2$	$[-100, 100]^D$
$f_7(\boldsymbol{x}) = \sum_{i=1}^{D} i x_i^4 + rand[0, 1)$	$[-1.28, 1.28]^D$
$f_8(\boldsymbol{x}) = \sum_{i=1}^{D} -x_i \sin \sqrt{\lvert x_i \rvert} + D \cdot 418.98288727243369$	$[-500, 500]^D$
$f_9(\boldsymbol{x}) = \sum_{i=1}^{D} \left[x_i^2 - 10 \cos(2\pi x_i) + 10 \right]$	$[-5.12, 5.12]^D$
$f_{10}(\boldsymbol{x}) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^{D} x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^{D} \cos(2\pi x_i) \right) + 20 + e$	$[-32, 32]^D$
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	$[-600, 600]^D$
$f_{12}(x) = \frac{\pi}{D} [10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 \{1 + 10 \sin^2(\pi y_{i+1})\} + (y_D - 1)^2] + \sum_{i=1}^{D} u(x_i, 10, 100, 4)$ where $y_i = 1 + \frac{1}{4}(x_i + 1)$ and $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \le x_i \le a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	$[-50, 50]^D$
$f_{13}(x) = 0.1[\sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 \{1 + \sin^2(3\pi x_{i+1})\} + (x_D - 1)^2 \{1 + \sin^2(2\pi x_D)\}] + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$	$[-50, 50]^D$

are selected according to [17]: Number of agents $N = 30$, $w = 0.729$, $c_1 = c_2 = 0.729 \times 2.05 = 1.49455$ and $V_{\max_j} = 0.5(u_j - l_j)$. The parameters of LPSO are: The number of sampling points $M = N$, $T_L = 200$, $N_{unimodal} = 5$, $N_{small} = 5$ for the lbest model and $N_{large} = N$ for the gbest model.

6.2 Experimental Results

The performance of three algorithms, gbest model PSO, lbest model PSO, and LPSO are compared. Table 2 shows the experimental results. The mean value and standard deviation of best objective values over 50 runs are shown in the top row for each function. The number of success runs, where the algorithm can find the near optimal value less than 10^{-7}, is shown in the bottom row. The best results among all algorithms are highlighted using bold face fonts.

The gbest model PSO attained the best results in unimodal functions f_1, f_2, f_3 and f_4. Also, the gbest model attained the best results in multimodal functions f_5, f_8 and f_9. It is thought that the gbest model is suitable not only to unimodal functions but also to functions where search points need to move a fairly long distance such as f_5 and f_8. The lbest model PSO attained the best results in multimodal functions f_{12} and f_{13}, and in the step function f_6.

It is thought that LPSO will show the intermediate performance between the gbest model and the lbest model. Nevertheless, LPSO attained the best results in multimodal functions f_{10} and f_{11}, the step function f_6 and the noisy function f_7. Thus, it is shown that dynamic selection of the gbest model and the lbest model can attain better result than pure gbest or lbest model.

LPSO got the first and second rank among three algorithms and did not get the worst rank in all functions. The average ranks of the gbest model, the lbest model and LPSO are 1.85, 2.42 and 1.73, respectively. LPSO attained the best performance as for the average rank.

The average success runs over 13 functions in the gbest model, the lbest model and LPSO are 21.00, 20.46 and 30.62, respectively. LPSO attained the best performance as for the average success runs.

Therefore, it is thought that LPSO showed the most stable performance.

As a reference, convergence graphs of test functions are shown in Figure 4, where the mean best objective values of LPSO, the gbest model PSO and the lbest model PSO are plotted over the number of function evaluations.

7. Conclusions

It is difficult to select a proper optimization strategy, because the proper strategy depends on the optimization problem and also on landscape currently being searched. In this study, in order to select a proper strategy of PSO dynamically, a dynamic selection of strategies is proposed where the gbest model is selected in unimodal landscape and the lbest model is selected in multimodal landscape. Various 13 functions are solved and the results are compared with those of the gbest and lbest models of PSO. It was shown that the proposed method sometimes outperformed the pure models and attained the most stable performance.

In the future, we will apply the dynamic selection of strategies to various algorithms. Also, we will apply the dynamic selection of algorithms such as an algorithm in unimodal landscape and another algorithm in multimodal landscape.

Acknowledgment

This research is supported in part by Grant-in-Aid for Scientific Research (C) (No. 22510166,24500177) of Japan society for the promotion of science and Hiroshima City University Grant for Special Academic Research (General Studies).

Table 2: Experimental results on standard PSOs and the proposed method. Mean value ± standard deviation and the number of success runs in 50 runs are shown.

	gbest model PSO	lbest model PSO	LPSO
f_1	**7.650e-118** ± **2.779e-117** [50]	3.392e-46 ± 7.533e-46 [50]	3.562e-109 ± 2.449e-108 [50]
f_2	**1.306e-39** ± **9.139e-39** [50]	4.722e-29 ± 3.509e-29 [0]	1.378e-38 ± 7.001e-38 [50]
f_3	**1.451e-13** ± **2.707e-13** [50]	1.912e+03 ± 9.675e+02 [0]	7.385e-13 ± 1.811e-12 [50]
f_4	**1.058e-06** ± **2.506e-06** [11]	1.496e-01 ± 8.532e-02 [0]	1.476e-06 ± 2.992e-06 [11]
f_5	**1.139e+01** ± **1.731e+01** [0]	7.128e+01 ± 4.073e+01 [0]	3.450e+01 ± 3.424e+01 [0]
f_6	2.900e+00 ± 6.275e+00 [19]	**0.000e+00** ± **0.000e+00** [50]	**0.000e+00** ± **0.000e+00** [50]
f_7	5.543e-03 ± 2.994e-03 [0]	1.047e-02 ± 3.555e-03 [0]	**4.201e-03** ± **1.557e-03** [0]
f_8	**3.043e+03** ± **6.714e+02** [0]	4.394e+03 ± 5.930e+02 [0]	4.313e+03 ± 5.978e+02 [0]
f_9	**7.245e+01** ± **1.612e+01** [0]	1.030e+02 ± 1.701e+01 [0]	7.466e+01 ± 1.933e+01 [0]
f_{10}	1.626e+00 ± 1.053e+00 [10]	1.581e-14 ± 4.884e-15 [50]	**1.105e-14** ± **5.012e-15** [50]
f_{11}	2.472e-02 ± 3.357e-02 [19]	3.149e-03 ± 9.259e-03 [16]	**6.191e-04** ± **1.844e-03** [39]
f_{12}	1.826e-01 ± 3.576e-01 [28]	**1.135e-21** ± **7.942e-21** [50]	4.147e-03 ± 2.031e-02 [48]
f_{13}	8.743e-02 ± 3.750e-01 [36]	**1.350e-32** ± **0.000e+00** [50]	1.352e-32 ± 1.726e-34 [50]

References

[1] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. of IEEE International Conference on Neural Networks*, vol. IV, Perth, Australia, 1995, pp. 1942–1948.

[2] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*. San Francisco: Morgan Kaufmann, 2001.

[3] R. Storn and K. Price, "Minimizing the real functions of the ICEC'96 contest by differential evolution," in *Proc. of the International Conference on Evolutionary Computation*, 1996, pp. 842–844.

[4] R. Storn and K. Price, "Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

[5] T. Takahama and S. Sakai, "Differential evolution with dynamic strategy and parameter selection by detecting landscape modality," in *Proc. of the 2012 IEEE Congress on Evolutionary Computation*, June 2012, pp. 2114–2121.

[6] T. Takahama and S. Sakai, "Large scale optimization by differential evolution with landscape modality detection and a diversity archive," in *Proc. of the 2012 IEEE Congress on Evolutionary Computation*, June 2012, pp. 2842–2849.

[7] S. Sakai and T. Takahama, "Large scale optimization by adaptive differential evolution with landscape modality detection and a diversity archive," *Journal of Business Studies*, vol. 58, no. 3, pp. 55–77, Mar. 2012.

[8] J. Liu and J. Lampinen, "A fuzzy adaptive differential evolution algorithm," *Soft Comput.*, vol. 9, no. 6, pp. 448–462, 2005.

[9] T. Takahama and S. Sakai, "Fuzzy c-means clustering and partition entropy for species-best strategy and search mode selection in nonlinear optimization by differential evolution," in *Proc. of the 2011 IEEE International Conference on Fuzzy Systems*, June 2011, pp. 290–297.

[10] J. Teo, "Exploring dynamic self-adaptive populations in differential evolution," *Soft Comput.*, vol. 10, no. 8, pp. 673–686, 2006.

[11] A. Qin, V. Huang, and P. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, april 2009.

[12] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, Oct. 2009.

[13] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 482–500, april 2012.

[14] Y.-W. Shang and Y.-H. Qiu, "A note on the extended Rosenbrock function," *Evolutionary Computation*, vol. 14, no. 1, pp. 119–126, 2006.

[15] X. Yao, Y. Liu, , and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, pp. 82–102, 1999.

[16] X. Yao, Y. Liu, K.-H. Liang, and G. Lin, "Fast evolutionary algorithms," in *Advances in Evolutionary Computing: Theory and Applications*, A. Ghosh and S. Tsutsui, Eds. New York, NY, USA: Springer-Verlag New York, Inc., 2003, pp. 45–94.

[17] R. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 1, 2001, pp. 81–86.

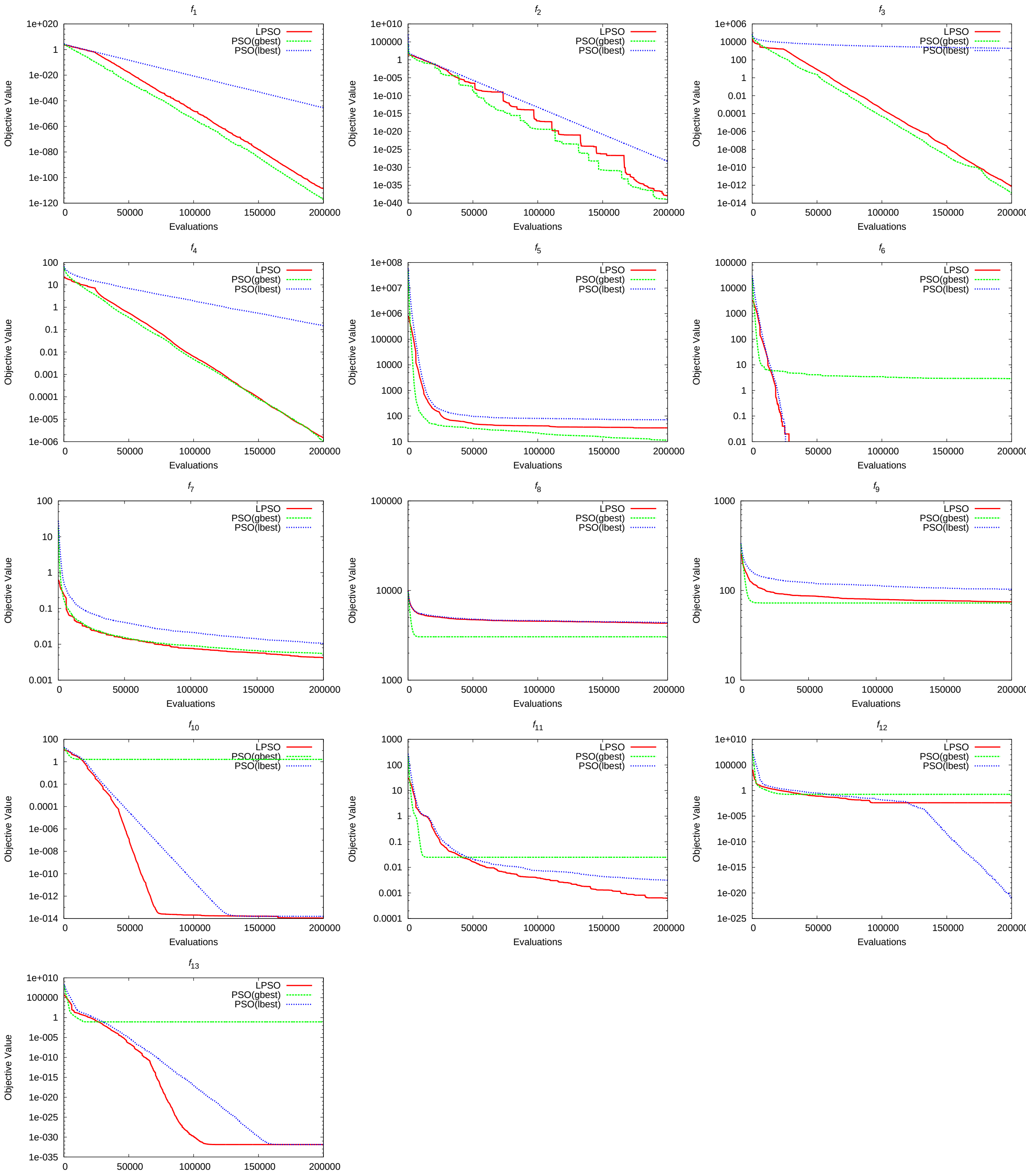

Fig. 4: Convergence graphs.