BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **18**, No 3
Sofia • 2018 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2018-0039

A Novel Multi-Epoch Particle Swarm Optimization Technique

Romasevych Yuriy, Loveikin Viatcheslav

Department of Machines and Equipment Design, National University of Life and Environmental Sciences of Ukraine, Kyiv, 03041, Ukraine
E-mails: romasevichyuriy@ukr.net lovvs@ukr.net

***Abstract*: *Since canonical PSO method has many disadvantages which do not allow to effectively reach the global minima of various functions it needs to be improved. The article refers to a novel Multi-Epoch Particle Swarm Optimization (ME-PSO) technique which has been developed by authors. ME-PSO algorithm is based on reinitializing of the stagnant swarm with low exploration efficiency. This approach provides a high rate of global best changing. As a result ME-PSO has great possibility of finding good local (or even global) optimum and does not trap in bad local optimum. In order to prove the advantages of the ME-PSO technique numerical experiments have been carried out with ten uni- and multimodal benchmark functions. Analysis of the obtained results convincingly showed significant superiority of ME-PSO over PSO and IA-PSO algorithms. It has been set that canonical PSO is a special case of ME-PSO.*

***Keywords*: *Particle swarm optimization, multi-epoch technique, Benchmark functions, convergence.*

1. Introduction

Bio-inspired optimization methods have great spread in many fields of the human activities [1]. The reasons are linked to their calculation advantages and implementation simplicity in different applications. One of the most powerful methods within such a class of methods is PSO [2]. The number of its applications is huge [2-6]. PSO and various modifications have been used for: learning and designing of artificial neural networks [2], calculations of various control problems [3, 4], signal processing [5], design [6], sentiment analysis [7], programming problems [8], etc. Almost in all of the referenced works were used modifications of the canonical PSO.

Indeed, many optimization problems have complications (stochastic influences [9], non-linearity [10, 11], multidimensionality and multi-extremal features [12], multi-objectivity [8, 13, 14], necessity to find the global extremum, etc.) which cause attempts of deep modification of PSO.

In this paper, we present a proposal, called Multi-Epoch Particle Swarm Optimization (ME-PSO). This method allows to improve significantly the exploration ability of a swarm and makes possible to use computation resources in a more effective manner.

2. Problem description

According to PSO, at the beginning of the search process every particle in the swarm has random position. Rather quickly PSO finds local optimum, after that better local optimum and so on and so forth. As the algorithm continues the successive local optima slightly differ and the quantity of iterations, required to reach further local optima, is extending. Hence, the swarm tends to stagnation and efficiency of the PSO respectively reduces. This problem is known as a premature convergence.

In order to overcome the premature convergence and improve PSO exploration ability a lot of its modifications have been proposed. Such modifications envisage various strategies: mutation [7], different topologies of the particles' connections [15] and topology variation [16], alteration of a swarm population [17], changing parameters of a swarm [18], varying the initial position and velocity of a swarm [19], adding extra terms in velocity expression or modification of canonical velocity formula [20], using many swarms in co-evolution interaction [21], integration PSO with other optimization methods [22], etc. Note that presented classification is not complete.

Some of the mentioned modifications have shown a good performance for test optimization problems. Nevertheless, there is a lack of the PSO modifications which allow to overcome the premature convergence in a simple manner. Here we mean the algorithms without high calculation complexity, the algorithms which are similar in simplicity to the canonical PSO. Hence, the further studies in the area of PSO-based techniques should be continued.

3. Proposed ME-PSO technique

3.1. Canonical PSO algorithm

In the PSO method, a swarm is a set of particles which move on the surface of a minimized function in order to find global minimum of the function. During its movement the particle improves the found minima and exchange information with neighbors. The position of the *i*-th particle is a set of its coordinates ($x_{i1}, x_{i2}, \ldots, x_{iD}$) in the search domain with dimensionality D.

At the initial stage of the PSO algorithm, the particles' positions are randomly initialized. Each particle is also described by the velocity vector, which is usually zero-vector for the initial iteration. During subsequent iterations, the components of the position and velocity vectors of a particle are being updated according to the formulas

$$\text{(1)} \qquad \begin{cases} \tilde{v}_{ij} = wv_{ij} + c_1r_1(p_{ij} - x_{ij}) + c_2r_2(g_j - x_{ij}), \\ \tilde{x}_{ij} = x_{ij} + \tilde{v}_{ij}, \end{cases}$$

where: $\tilde{v}_{ij}$ and $\tilde{x}_{ij}$ are the new *j*-th component of velocity and position vectors of the *i*-th particle; p_{ij} is the best position, that has been found by the *i*-th particle on the previous iterations (personal best); g_j is the best position, that has been found by the whole swarm on the previous iterations (global best); w is the inertial coefficient; c_1 and c_2 are cognitive and social coefficients respectively; r_1, r_2 are random numbers that are generated on the interval [0, 1]. The inertial coefficient w determines the influence of the previous velocity of the particle to the $\tilde{v}_{ij}$. The value of the cognitive coefficient c_1 characterizes the degree of individual particle behavior, its "desire" to move towards personal best. The value of the social coefficient c_2 reflects the degree of collective behavior, the "desire" to move towards global best.

For the very first iteration the initial positions of the particles are considered as the best.

An iteration of PSO algorithm includes applying the formulas (1) and updating the of p_{ij} and g_j values according to the rules:

$$\begin{cases} p_j = x_j & \text{if} \quad f(x_j) < f(p_j), \\ g = p_j & \text{if} \quad f(p_j) < f(g), \end{cases} \tag{2}$$

where f is a function to be minimized.

PSO execution provides advantageous exploration of the minimized function. For simple functions PSO commonly finds global minimum. As for topologically complicated functions PSO algorithm finds bad local minimum. This disadvantage is caused by premature convergence of the algorithm. In the following we propose a simple modifocation of PSO which eliminates mentioned disadvantage.

3.2. Essence of the ME-PSO technique

Proposed ME-PSO algorithm is based on the monitoring of the global optimum search performance (herein, for clarity sake, we will refer as minimum of a function). The main idea of the novel technique is the following: if the rate of the swarm global best reducing is low, then all the swarm particles positions must be reinitialized in a random way (the new epoch of the swarm commences). The global best of the swarm (in the new epoch) for the first iteration is the same as it was for the last iteration of the swarm in the previous epoch.

Moving on the surface of a function a particle may trap in a minimum that would be better than the previous epoch global best. It should be noted that for some number of iterations (just after reinitialization of the swarm) the particles move without any improvement of the global best. Soon a particle may find the local minimum that is better than the current global best.

We should set a criterion of swarm stagnation. Such criterion may be Global Best (GB) reduction rate described by the following expression:

$$R = \frac{\mathrm{GB}_i - \mathrm{GB}_{i-1}}{\mathrm{GB}_i}, \tag{3}$$

where GB_i and GB_{i-1} are global bests of a swarm for the i-th (current) and (i – 1)-th (previous) iterations. Equation (3) shows how much the global best of a swarm reduces during an iteration. Thus, value of R must be calculated at the end of every iteration. If the global best reduction rate is low then the swarm must be reinitialized. The condition of the swarm reinitialization is

(4) $$\text{AR} \geq R,$$
where AR is Acceptable Rate of the global best reduction (this value must be set by ME-PSO user).

The value of the AR must be set in view of the recommendations:

- big value of AR causes frequent reinitialization of the swarm and, as a result, during many iterations particles move without improving the global best;
- small value of AR leads to jamming of the particles in local minima and stagnation of the swarm (note, if the AR=0 ME-PSO reduces to canonical PSO).

The issue of assigning AR value is still open for discussions and it is necessary to investigate it in the further studies. In the following we set AR=0.01. Such value of AR provides quite good balance between function exploration and reinitialization.

Other criteria, which may be used for the swarm reinitialization, are presented in the Table 1. They should be checked at the end of the each iteration as well.

Table 1. Possible criteria which may be used as condition of a swarm reinitialaization

Formula	Description
$\text{GB}_i = \text{GB}_{i-q},\ q = 1, ..., Q,$ where Q is the number of the iterations which must be set before run of the algorithm	If the value of the global best has not been changed during Q iterations this may mean that the particles have been trapped in local minima and they cannot leave it. The number Q may be set as a fraction of the total number of iterations N. For instance, Q=N(0.01…0.10)
$\frac{1}{E}\sum_{e=1}^{E}\text{LB}_i^e \approx (0.99...0.90)\text{GB}_i,$ where E is some quantity of particles which is lesser than the swarm population; LB is the Local Bests of the particles. Subscript means the i-th iteration, superscript means a number of the particle	If the E particles are close to the best particle in a swarm that mean they have a little chances to find the minimum that would be better than the current global best. In order to avoid swarm stagnation about the global best it is necessary to reinitialize the swarm. The number E must be set as an integer, for example, E=SP(0.1…0.5), where SP is the Swarm Population

In the article we use only (1) and (2) expressions as condition of the swarm reinitialization. Note, all or just only one of the described criteria (Table 1) may be used for this purpose.

In order to provide the high search ability of particles we propose to eliminate their inertial feature. This requires to set inertia coefficient equal to zero w=0. It causes rapid movement of particles on a function surface and, as a result, the bigger area of the function domain may be explored.

The novel ME-PSO technique can be clarified within the support of the following pseudocode:

Set the parameters c_1, c_2, SP, stop criterion (number of iterations, cost function value, etc.) and AR;
Initialize particles positions and velocities;
Calculate global best;
    Do
        Update the particles positions and velocities;
        For each particle check the excess search domain condition;

Calculate personal bests and global best;
Calculate R;
If $R \leq$AR **then**
Reinitialize particles positions and velocities;
Until stop condition is met.

During initialization and reinitialization, all components of particles' positions should be set as random numbers in the search domain and all components of a particles' velocity should be set equal to zero. The approach described allows to use computational resources more efficiently. Further study will make possible to establish how ME-PSO copes with the different optimization problems.

3.3. Experiment

In order to show advantages of the ME-PSO numerical experiments have been performed. We choose ten benchmark functions: uni- and multimodal (Table 2). All of the chosen functions have different topology features but each function has global minima which are equal to zero.

Table 2. Ten benchmark function for numerical experiment

Benchmark function	Formula	Search domain
Spherical	$f1 = \sum_{i=1}^{D} x_i^2$	$-20 \leq x_i \leq 20$
Elliptical	$f2 = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} x_i^2$	$-2 \leq x_i \leq 2$
Schwefel No 1	$f3 = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j \right)^2$	$-10 \leq x_i \leq 10$
Rosenbrock	$f4 = \sum_{i=1}^{D-1} \left(100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \right)$	$-10 \leq x_i \leq 10$
Rastrigin	$f5 = \sum_{i=1}^{D} \left(x_i^2 - 10\cos(2\pi x_i) + 10 \right)$	$-5 \leq x_i \leq 5$
Griewank	$f6 = 4000^{-1} \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} x_i i^{-0.5} + 1$	$-100 \leq x_i \leq 100$
Alpine	$f7 = \sum_{i=1}^{D} \lvert x_i \sin(x_i) + 0.1 x_i \rvert$	$-10 \leq x_i \leq 10$
Schwefel No 2	$f8 = D^{-1} \sum_{i=1}^{D} \left(x_i \sin(\lvert x_i \rvert^{-0.5}) \right) + 418.983$	$-500 \leq x_i \leq 500$
Ackley	$f9 = -20\exp\left(-0{,}2 \left(D^{-1} \sum_{i=1}^{D} x_i^2 \right)^{0.5} \right) - \exp\left(D^{-1} \sum_{i=1}^{D} \cos(2\pi x_i) \right) + 20 + e$	$-30 \leq x_i \leq 30$
Weierstrass	$f10 = D^{-1} \sum_{i=1}^{D} \sum_{k=0}^{20} \left(0.5^k \cos(2\pi 3^k (x_i + 0.5)) \right) - \sum_{k=0}^{20} \left(0.5^k \cos(\pi 3^k) \right)$	$-0.5 \leq x_i \leq 0.5$

In order to establish how benchmark functions Dimensions (*D*) influences to ME-PSO performance, experiments were carried out for numbers of *D*: 10, 30, 50 and 200 (for the last experimental series). As indicators of the algorithms efficiency average and median values were used. Standard Deviation (SD) indicates dispersion of the reached minima in relation to average values. All calculations were carried out for PSO, IA-PSO [16] and ME-PSO techniques. Comparison of different approaches, which are implemented in the ME-PSO and IA-PSO algorithms, will give the information about the efficiency of overcoming the premature convergence.

In all experiments the number of iterations *N* as a stop criterion was used.

In order to obtained proper statistical results each numerical experiment has been run 100 times. In each run the particles' positions were random. Parameters of the swarm were the same for all experiments (Table 3).

Table 3. Swarm parameters for all numerical experiments

Parameters of the swarm	Value
c_1	1.19
c_2	1.19
w	0.72
Swarm population	30
Connection topology	Full

4. Results and discussion

Results of the first series of experiments (*N*=250) allow to determine the algorithms' performance at early stages of the exploration (Table 4).

In Table 4 and further tables the best values of average and median are in bold.

Table 4 shows that on early stage of exploration ME-PSO has reached not good minima values for almost all benchmark functions. The only one exception is the function *f*5. The best performance in this experimental series relates to IA-PSO technique. For *D*=10 it allows to find local minima which are very close (in topology sense) to the global minimum of the functions *f*1, *f*2, *f*3, *f*10.

The worst results ME-PSO has shown for the functions *f*4 and *f*8. It is caused by small number of iterations. Thus, there is a need to study how ME-PSO works with big number of *N*. It was the purpose of the second series of experiment in which *N*=5000 (Table 5). The calculations were carried out for the most difficult functions to minimize.

Comparison of data in Table 4 and Table 5 supports the statement that an increasing of the iteration number *N* makes it possible to reduce the average and median values of reached minima. Moreover, for some cases ME-PSO has reached almost computer zero (function *f*5 with *D*=10).

Table 4. Results of the first experimental series

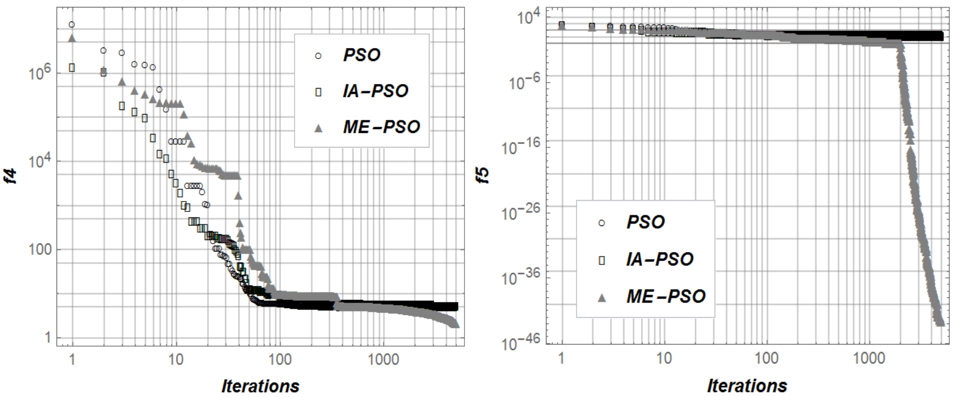
Functions	PSO			IA-PSO			ME-PSO		
	Average	Median	SD	Average	Median	SD	Average	Median	SD
D=10									
$f1$	1.11×10^{-17}	2.26×10^{-18}	2.78×10^{-17}	**4.09×10^{-30}**	**4.89×10^{-32}**	1.69×10^{-29}	1.02×10^{-7}	4.02×10^{-8}	1.70×10^{-7}
$f2$	8.69×10^{1}	1.30×10^{-10}	8.62×10^{2}	**1.11×10^{-30}**	**3.98×10^{-33}**	1.01×10^{-29}	5.00×10^{-1}	4.26×10^{-1}	4.52×10^{-1}
$f3$	2.14×10^{-6}	1.69×10^{-8}	9.85×10^{-6}	**7.21×10^{-11}**	**1.75×10^{-14}**	4.32×10^{-10}	2.20×10^{-2}	1.42×10^{-2}	2.44×10^{-2}
$f4$	1.16×10^{1}	4.59×10^{0}	3.37×10^{1}	**7.35×10^{0}**	**7.12×10^{0}**	9.39×10^{-1}	2.04×10^{1}	7.63×10^{0}	3.10×10^{1}
$f5$	1.42×10^{1}	1.29×10^{1}	6.42×10^{0}	1.58×10^{1}	1.47×10^{1}	6.77×10^{0}	**5.68×10^{0}**	**5.05×10^{0}**	2.27×10^{0}
$f6$	**1.01×10^{-1}**	**8.73×10^{-2}**	6.53×10^{-2}	2.57×10^{-1}	2.39×10^{-1}	1.41×10^{-1}	1.69×10^{-1}	1.43×10^{-1}	1.00×10^{-1}
$f7$	**2.41×10^{-3}**	**1.67×10^{-5}**	1.57×10^{-3}	1.09×10^{0}	7.33×10^{-1}	1.07×10^{0}	3.37×10^{-3}	9.07×10^{-4}	7.58×10^{-3}
$f8$	**4.03×10^{2}**	**4.03×10^{2}**	1.86×10^{0}	4.10×10^{2}	4.10×10^{2}	1.13×10^{0}	4.13×10^{2}	4.12×10^{2}	3.71×10^{-1}
$f9$	3.00×10^{-1}	**4.73×10^{-9}**	5.98×10^{-1}	1.60×10^{0}	2.04×10^{0}	1.33×10^{0}	**2.05×10^{-3}**	6.58×10^{-4}	7.25×10^{-3}
$f10$	8.40×10^{-2}	4.47×10^{-2}	9.51×10^{-2}	**7.90×10^{-3}**	**1.40×10^{-16}**	3.89×10^{-2}	8.30×10^{-3}	1.47×10^{-3}	2.31×10^{-2}
D=30									
$f1$	1.87×10^{0}	4.08×10^{-1}	4.34×10^{0}	**5.19×10^{-8}**	**8.25×10^{-9}**	3.37×10^{-7}	7.94×10^{-2}	6.50×10^{-2}	5.23×10^{-2}
$f2$	7.86×10^{2}	3.35×10^{2}	1.20×10^{2}	**4.35×10^{-11}**	**5.53×10^{-12}**	1.35×10^{-10}	8.44×10^{1}	8.14×10^{1}	3.74×10^{1}
$f3$	3.89×10^{1}	3.40×10^{1}	2.37×10^{1}	**3.62×10^{-1}**	**2.54×10^{-1}**	3.42×10^{-1}	1.46×10^{1}	1.44×10^{1}	4.10×10^{0}
$f4$	4.78×10^{2}	2.37×10^{2}	6.84×10^{2}	**2.83×10^{1}**	**2.86×10^{1}**	1.14×10^{0}	1.47×10^{2}	1.36×10^{2}	7.52×10^{1}
$f5$	9.73×10^{1}	9.84×10^{1}	2.25×10^{1}	1.11×10^{2}	1.09×10^{2}	2.64×10^{1}	**5.71×10^{1}**	**5.77×10^{1}**	1.24×10^{1}
$f6$	3.67×10^{-1}	3.15×10^{-1}	2.58×10^{-1}	**3.36×10^{-2}**	**5.18×10^{-8}**	3.88×10^{-2}	3.05×10^{-1}	2.35×10^{-1}	2.03×10^{-1}
$f7$	1.89×10^{0}	1.67×10^{0}	1.25×10^{0}	1.23×10^{1}	1.24×10^{1}	3.73×10^{0}	**8.04×10^{-1}**	**6.49×10^{-1}**	5.37×10^{-1}
$f8$	**4.03×10^{2}**	**4.03×10^{2}**	2.01×10^{0}	4.13×10^{2}	4.13×10^{2}	1.38×10^{0}	4.10×10^{2}	4.10×10^{2}	3.09×10^{-1}
$f9$	6.73×10^{0}	6.45×10^{0}	1.93×10^{0}	3.18×10^{0}	3.47×10^{0}	1.01×10^{0}	**2.10×10^{0}**	**2.10×10^{0}**	5.26×10^{-1}
$f10$	5.08×10^{-1}	5.04×10^{-1}	9.92×10^{-2}	5.02×10^{3}	2.07×10^{4}	4.36×10^{-2}	**2.09×10^{-1}**	**2.07×10^{-1}**	6.06×10^{-2}
D=50									
$f1$	7.08×10^{1}	5.98×10^{1}	4.63×10^{-1}	**2.01×10^{-4}**	**1.76×10^{-4}**	1.42×10^{-4}	2.28×10^{0}	2.17×10^{0}	7.99×10^{-1}
$f2$	7.77×10^{3}	4.99×10^{3}	8.73×10^{3}	**7.58×10^{-6}**	**2.88×10^{-6}**	3.03×10^{-5}	6.16×10^{2}	6.01×10^{2}	2.15×10^{2}
$f3$	7.08×10^{2}	6.47×10^{2}	2.79×10^{2}	**1.18×10^{1}**	**1.02×10^{1}**	6.50×10^{0}	7.25×10^{1}	7.22×10^{1}	1.73×10^{1}
$f4$	2.00×10^{4}	1.96×10^{4}	5.75×10^{2}	**4.85×10^{1}**	**4.85×10^{1}**	5.18×10^{-1}	4.72×10^{2}	4.47×10^{2}	1.52×10^{2}
$f5$	3.46×10^{2}	3.52×10^{2}	6.75×10^{1}	2.49×10^{2}	2.46×10^{2}	4.31×10^{1}	**1.31×10^{2}**	**1.28×10^{2}**	2.55×10^{1}
$f6$	1.43×10^{0}	1.39×10^{0}	2.37×10^{-1}	**5.63×10^{-2}**	**1.69×10^{-4}**	7.52×10^{-2}	1.04×10^{0}	1.04×10^{0}	2.02×10^{-2}
$f7$	1.16×10^{1}	1.16×10^{1}	3.95×10^{0}	2.79×10^{1}	2.77×10^{1}	6.33×10^{0}	**5.51×10^{0}**	**5.12×10^{0}**	2.09×10^{0}
$f8$	**4.05×10^{2}**	**4.05×10^{2}**	1.05×10^{-1}	4.16×10^{2}	4.16×10^{2}	5.00×10^{-1}	4.15×10^{2}	4.15×10^{2}	7.01×10^{-2}
$f9$	1.70×10^{1}	1.11×10^{1}	3.84×10^{0}	**3.21×10^{0}**	3.90×10^{0}	1.47×10^{0}	3.77×10^{0}	**3.71×10^{0}**	4.69×10^{-1}
$f10$	7.01×10^{-1}	6.97×10^{-1}	8.45×10^{-2}	**7.87×10^{-3}**	**4.07×10^{-3}**	1.21×10^{-2}	3.51×10^{-1}	3.58×10^{-1}	5.69×10^{-1}

Almost all results that are related to ME-PSO are better than those that were obtained with PSO and IA-PSO techniques. The exceptions are functions $f3$, $f4$ and $f8$. IA-PSO finds good local minimum of the $f3$ (with D=50) quite rapid. However, complicated topology of function *f*4 causes the premature convergence both PSO and IA-PSO, especially for *D*=50.

Table 5. Results of the second experimental series

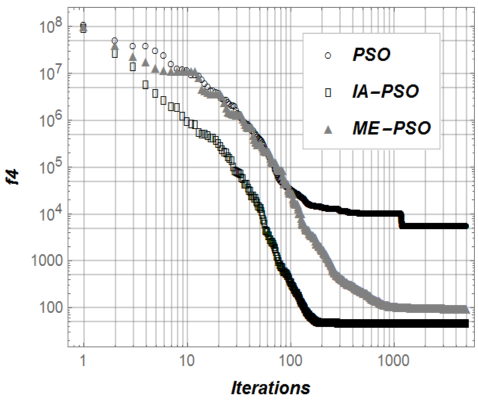
Functions	PSO			IA-PSO			ME-PSO		
	Average	Median	SD	Average	Median	SD	Average	Median	SD
D=10									
*f*4	5.18×10^{0}	4.33×10^{0}	6.56×10^{0}	6.43×10^{0}	6.22×10^{0}	1.13×10^{0}	$\mathbf{2.18\times10^{0}}$	$\mathbf{1.84\times10^{0}}$	1.89×10^{0}
*f*5	1.14×10^{1}	1.04×10^{1}	5.54×10^{0}	7.84×10^{0}	7.33×10^{0}	4.55×10^{0}	$\mathbf{8.55\times10^{-37}}$	$\mathbf{1.97\times10^{-47}}$	5.48×10^{-36}
*f*8	4.15×10^{2}	4.15×10^{2}	2.19×10^{-1}	4.15×10^{2}	4.15×10^{2}	2.11×10^{-1}	$\mathbf{4.03\times10^{2}}$	$\mathbf{4.03\times10^{2}}$	1.33×10^{0}
D=30									
*f*4	2.08×10^{2}	1.71×10^{2}	1.56×10^{2}	$\mathbf{2.78\times10^{1}}$	2.79×10^{1}	1.00×10^{0}	5.06×10^{1}	$\mathbf{2.72\times10^{1}}$	3.53×10^{1}
*f*5	5.13×10^{1}	4.93×10^{1}	1.77×10^{1}	4.62×10^{1}	4.64×10^{1}	1.58×10^{1}	$\mathbf{9.58\times10^{0}}$	$\mathbf{9.95\times10^{0}}$	2.45×10^{0}
*f*8	$\mathbf{4.03\times10^{2}}$	$\mathbf{4.03\times10^{2}}$	4.32×10^{-1}	4.16×10^{2}	4.16×10^{2}	2.34×10^{-1}	4.15×10^{2}	4.15×10^{2}	1.61×10^{-11}
*f*9	3.47×10^{0}	3.28×10^{0}	9.32×10^{-1}	2.58×10^{0}	2.94×10^{0}	1.03×10^{0}	$\mathbf{6.39\times10^{-12}}$	$\mathbf{4.62\times10^{-12}}$	6.28×10^{-12}
D=50									
*f*3	2.19×10^{2}	2.16×10^{2}	6.35×10^{1}	$\mathbf{3.73\times10^{-11}}$	$\mathbf{1.06\times10^{-14}}$	3.14×10^{-10}	2.29×10^{0}	2.20×10^{0}	6.37×10^{-1}
*f*4	5.35×10^{3}	3.37×10^{3}	5.16×10^{3}	$\mathbf{4.79\times10^{1}}$	$\mathbf{4.85\times10^{1}}$	9.86×10^{-1}	9.02×10^{1}	9.48×10^{1}	4.08×10^{1}
*f*5	1.94×10^{2}	1.92×10^{2}	3.94×10^{1}	2.50×10^{2}	2.54×10^{2}	4.30×10^{1}	$\mathbf{4.30\times10^{1}}$	$\mathbf{4.31\times10^{1}}$	7.91×10^{0}
*f*7	1.90×10^{1}	1.89×10^{1}	5.82×10^{0}	1.07×10^{1}	1.03×10^{1}	4.50×10^{0}	$\mathbf{2.59\times10^{-2}}$	$\mathbf{1.80\times10^{-2}}$	2.57×10^{-2}
*f*8	$\mathbf{4.15\times10^{2}}$	$\mathbf{4.15\times10^{2}}$	2.25×10^{-1}	4.16×10^{2}	4.16×10^{2}	1.87×10^{-1}	$\mathbf{4.15\times10^{2}}$	$\mathbf{4.15\times10^{2}}$	1.07×10^{-2}
*f*9	1.16×10^{1}	1.17×10^{1}	1.62×10^{0}	2.00×10^{0}	2.00×10^{0}	1.86×10^{-3}	$\mathbf{6.78\times10^{-8}}$	$\mathbf{5.36\times10^{-8}}$	4.65×10^{-8}

In order to compare PSO, IA-PSO and ME-PSO performances the graphs have been built (Fig. 1 and Fig. 2). Vertical and horizontal axes of the following graphs are presented in logarithmic scale.

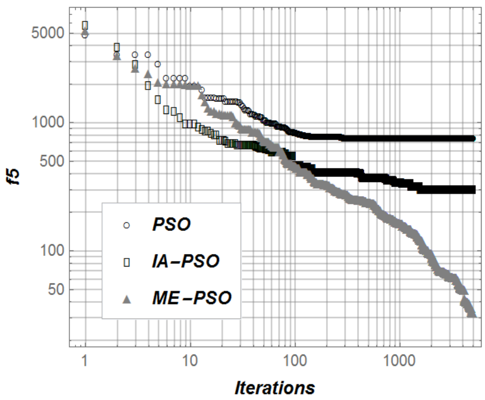


a) *f*4 b) *f*5

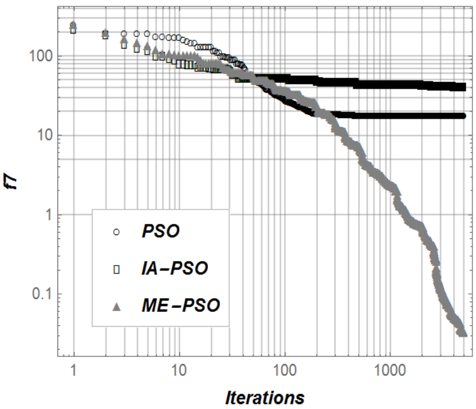
Fig. 1. PSO, IA-PSO and ME-PSO performances during minimizing of the benchmark functions *f*4 and *f*5 (*D*=10)



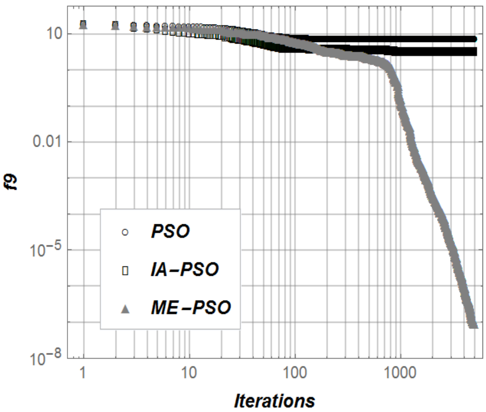
a) *f*4



b) *f*5



c) *f*7



d) *f*9

Fig. 2. PSO, IA-PSO and ME-PSO performances during minimizing of the benchmark functions *f*4, *f*5, *f*7 and *f*9 (*D*=50)

The graphs on Fig. 2 (b, c, d) make it obvious that ME-PSO has no premature convergence: the algorithm execution provides reduction of the global best during all iterations.

This is the biggest difference between ME-PSO and algorithms PSO and IA-PSO. Graphs on the Fig. 1 and Fig. 2 clearly show that PSO and IA-PSO converge rather quickly. In contrast, ME-PSO continues to minimize almost all benchmark functions during all iterations.

In order to support that suggestion the third series of experiment was performed. All the calculations was carried out with the *f*4 and *f*5 functions with *D*=50.

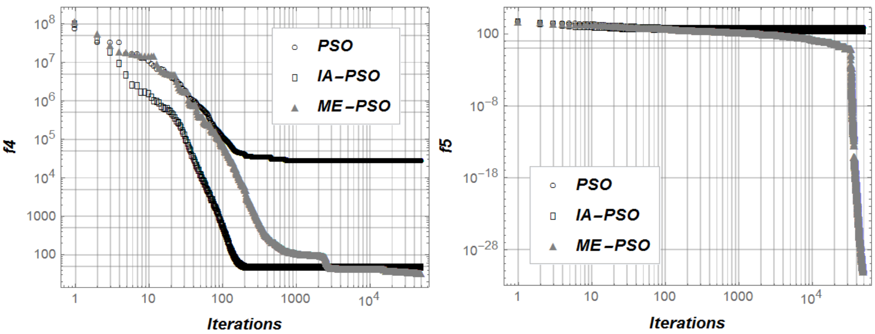
We choose *f*4 because IA-PSO has better than ME-PSO performance for its minimization. In order to investigate the impact of iterations number *N* on the IA-PSO and ME-PSO performances we set *N*=50,000.

The choice of *f*5 for third experimental series is caused by the fact of bad efficiency of ME-PSO for that function on the previous experimental series. All the figures were set to the Table 6.

Table 6. Results of the third experimental series

Func-tions	PSO			IA-PSO			ME-PSO		
	Average	Median	SD	Average	Median	SD	Average	Median	SD
*f*4	1.98×10^{4}	1.94×10^{4}	5.61×10^{2}	4.72×10^{1}	4.88×10^{1}	8.99×10^{-1}	$\mathbf{3.62\times10^{1}}$	$\mathbf{3.79\times10^{1}}$	2.02×10^{0}
*f*5	3.44×10^{2}	3.43×10^{2}	6.66×10^{1}	2.44×10^{2}	2.51×10^{2}	5.301×10^{1}	$\mathbf{3.34\times10^{-35}}$	$\mathbf{2.50\times10^{-36}}$	8.32×10^{-37}

Obtained results show that advantages of ME-PSO (especially for complicated functions) are revealed at big number of iterations. The graphs on Fig. 3 show the bigger *N* the better ME-PSO performance.



a) *f*4 b) *f*5

Fig. 3. PSO, IA-PSO and ME-PSO performances during minimizing of the benchmark functions *f*4 and *f*5 (D=50)

The most difficult benchmark function for all algorithms is *f*8. None of them have found a good solution. At early stages of the exploration these algorithms find bad local minima of *f*8. Even reinitialization of a swarm does not solve the problem: all the particles in a new epoch swarm have a great tendency to move toward previous global best. They have no time for proper exploration of the *f*8. This ME-PSO weakness (only for some of the complicated optimization problems) causes the necessity for further improving of the proposed algorithm.

One of the possible ways to solve that problem is varying parameter *AR* during optimization process. For instance, AR can be a function of the current global best or current iteration. That issue is a matter for further investigations.

Some optimization algorithms fail when the dimension of the cost function is more than 100. That is why the fourth series of experiment was conducted under condition D=200. Two benchmark functions (*f*4 and *f*7) were chosen for that series. They have different features: the first one is unimodal and the second one is multimodal. For these high-dimensional problems we set N=50,000. All the obtained figures are in Table 7.

Table 7. Results of the fourth experimental series

Func-tions	PSO			IA-PSO			ME-PSO		
	Average	Median	SD	Average	Median	SD	Average	Median	SD
*f*4	1.46×10^{6}	1.42×10^{6}	2.09×10^{4}	1.98×10^{2}	1.97×10^{2}	7.90×10^{0}	$\mathbf{1.96\times10^{2}}$	$\mathbf{1.94\times10^{2}}$	5.04×10^{0}
*f*7	1.92×10^{2}	1.83×10^{2}	4.90×10^{1}	7.90×10^{1}	7.33×10^{1}	1.01×10^{1}	$\mathbf{1.38\times10^{1}}$	$\mathbf{1.33\times10^{1}}$	4.22×10^{0}

Graphs which related to PSO, IA-PSO and ME-PSO performances for the high-dimensional optimization problems are shown on Fig 4.

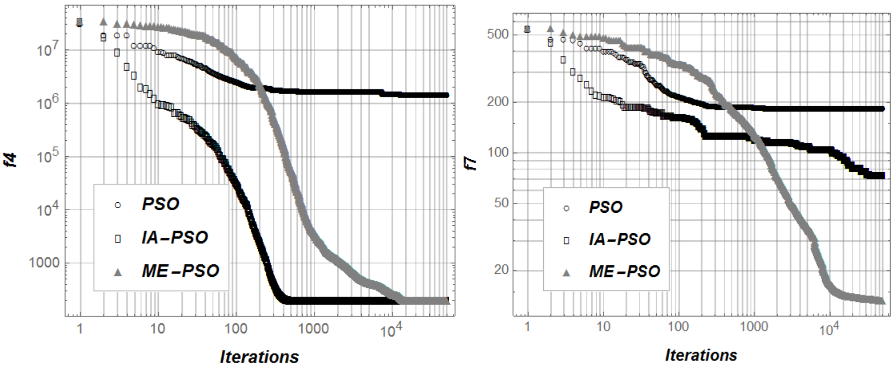


Fig. 4. PSO, IA-PSO and ME-PSO performances during minimizing of the benchmark functions *f*4 and *f*7 (*D*=200)

Data in Table 7 and graphs on Fig. 4 clearly prove the superiority of ME-PSO. Although for *f*4 the difference between IA-PSO and ME-PSO is slight.

Fig. 4a shows the convergence of all algorithms. It is an obstacle for further function minimization. In order to prevent it further improvement of ME-PSO should be carried out. The ultimate goal is to find a PSO modification which is invariant to problem dimensionality and has high exploration abilities.

5. Conclusion

In the article we proposed the novel PSO-based technique (ME-PSO). The basic idea of it is in reinitialization of the stagnant swarm. The article contains the description of the stagnation criteria and one of them has been used in the calculations presented above. The criterion used in calculations appeals to the rate of global best reduction. If it is low then a swarm should be reinitialized (the new epoch of swarm is commencing).

The value of Acceptable Rate (AR) is a matter for further studies. It is necessary to found the connections between AR and parameters of optimization problem: its dimensionality, search domain, a function topology, etc.

The main advantage of the ME-PSO is the following: the greater the number of iterations the better the value of a reached extremum of a function. Proposed ME-PSO algorithm may be combined with other PSO modifications, which have been mentioned in the section „Problem description”. The reasonable combinations of ME-PSO and other PSO-based techniques, impact of parameters on the algorithm performance are the issues for further studies.

References

1. B o z o r g - H a d d a d, O., M. S o l g i, H. A. L o á i c i g a. Meta-Heuristic and Evolutionary Algorithms for Engineering Optimization. Hoboken, USA, John Wiley & Sons Inc, 2017.
2. K e n n e d y, J., R. C. E b e r h a r t. Particle Swarm Optimization. – In: Proc. of IEEE International Conference on Neural Networks, 1995, pp. 1942-1948.
3. K i r a n y a z, S., T. I n c e, A. Y i l d i r i m, M. G a b b o u j. Evolutionary Artificial Neural Networks by Multi-Dimensional Particle Swarm Optimization. – Neural Networks, Vol. **22**, 2009, Issue 10, pp. 1448-1462.
4. H e o, J. S., K. Y. L e e, R. G a r d u n o-R a m i r e z. Multiobjective Control of Power Plants Using Particle Swarm Optimization Techniques. – IEEE Transactions on Energy Conversion, Vol. **21**, 2006, Issue 10, pp. 552-561.
5. Z a m a n i, M., M. K a r i m i-G h a r t e m a n i, N. S a d a t i, M. P a r n i a n i. Design of a Fractional Order PID Controller for an AVR Using Particle Swarm Optimization. – Control Engineering Practice, Vol. **17**, 2009, Issue 12, pp. 1380-1387.
6. C h a n d e r, A., A. C h a t t e r j e e, P. S i a r r y. A New Social and Momentum Component Adaptive PSO Algorithm for Image Segmentation. – Expert Systems with Applications, Vol. **38**, 2011, Issue 5, pp. 4998-5004.
7. B o r d b a r , S., P. S h a m s i n e j a d. A New Opinion Mining Method Based on Fuzzy Classifier and Particle Swarm Optimization (PSO) Algorithm. – Cybernetics and Information Technologies, Vol. **18**, 2018, No 2, pp. 36-50.
8. H e, Q., Y. L v. Particle Swarm Optimization Based on Smoothing Approach for Solving a Class of Bi-Level Multiobjective Programming Problem. – Cybernetics and Information Technologies, Vol. **17**, 2017, No 3, pp. 59-74.
9. B a o, G. Q., D. Z h a n g, J. H. S h i, J. Z. J i a n g. Optimal Design for Cogging Torque Reduction of Transverse Flux Permanent Motor Using Particle Swarm Optimization Algorithm. – Power Electronics and Motion Control Conference, Vol. **4**, 2004.
10. S h u r u b, Y. V., A. O. D u d n y k, D. S. L a v i n s k i y. Optimization of Regulators of Frequency Controlled Induction Electric Drives under the Stochastic Loadings. – Journal Tekhnichna Elektrodynamika, Vol. **4**, 2016, pp. 53-55.
11. T a h e r, N. A New Fuzzy Adaptive Hybrid Particle Swarm Optimization Algorithm for Non-Linear, Non-Smooth and Non-Convex Economic Dispatch Problem. – Applied Energy, Vol. **87**, 2010, Issue 1, pp. 327-339.
12. R a o, R. V., V. J. S a v s a n i, D. P. V a k h a r i a. Teaching-Learning-Based Optimization: An Optimization Method for Continuous Non-Linear Large Scale Problems. – Information Sciences, Vol. **183**, 2012, Issue 1, pp. 1-15.
13. L i u, D., K. C. T a n, C. K. G o h, W. K. H o. A Multiobjective Memetic Algorithm Based on Particle Swarm Optimization. – IEEE Transactions on Systems Man and Cybernetics, Part B (Cybernetics), Vol. **37**, 2007, Issue 1, pp. 42-50.
14. L o v e i k i n, V. S., Y. O. R o m a s e v y c h. Dynamic Optimization of a Mine Winder Acceleration Mod. – Naukovyi Visnyk Natsionalnoho Hirnychoho Universytetu, Vol. **4**, 2017, pp. 55-61.
15. C l e r c, M. Back to random topology [Electronic resource]. **http://clerc.maurice.free.fr/pso/random_topology.pdf**
16. R i c h a r d s, M., D. V e n t u r a. Dynamic Sociometry in Particle Swarm Optimization. – In: Proc. of Joint Conference of Information Sciences, 2003, pp. 1557-1560.
17. C l e r c, M. Variable PSO [Electronic resource]. **http://clerc.maurice.free.fr/pso/2011-01-20_Variable_PSO.zip**
18. S u r e s h, K., S. G h o s h, D. K u n d u, A. S e n. Inertia-Adaptive Particle Swarm Optimizer for Improved Global Search. – Intelligent Systems Design and Applications, 2008.
19. Y o n g, D., W. W u C h u a n s h e n g, G. H a i m i n. Particle Swarm Optimization Algorithm with Adaptive Chaos Perturbation. – Cybernetics and Information Technologies, Special Issue on Logistics, Informatics and Service Science, Vol. **15**, 2015, No 6, pp. 70-80.

20. J o r d a n, J., S. H e l w i n g, R. W a n k a. Social Interaction in Particle Swarm Optimization, the Ranked FIPS, and Adaptive Multi-Swarms. – In: Proc. of 10th Annual Conference on Genetic and Evolutionary Computation, 2008, pp. 49-56.
21. P a r s o p o u l o s, K. E. Parallel Cooperative Micro-Particle Swarm Optimization: A Master-Slave Model. – Applied Soft Computing, Vol. **12**, 2012, Issue 11, pp. 3552-3579.
22. G a r g, H. A Hybrid PSO-GA Algorithm for Constrained Optimization Problems. – Applied Mathematics and Computation, Vol. **274**, 2016, pp. 292-305.

Received 22.06.2018; Second Version 12.08.2018; Accepted 17.08.2018