

Research Article

The Inertia Weight Updating Strategies in Particle Swarm Optimisation Based on the Beta Distribution

Petr Maca and Pavel Pech

*Department of Water Resources and Environmental Modeling, Faculty of Environmental Sciences,
Czech University of Life Sciences Prague, Kamýcka 1176, Praha 6, 165 21 Suchbát, Czech Republic*

Correspondence should be addressed to Petr Maca; maca@fzp.czu.cz

Received 6 June 2014; Revised 21 August 2014; Accepted 8 September 2014

Academic Editor: Fang Zong

Copyright © 2015 P. Maca and P. Pech. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The presented paper deals with the comparison of selected random updating strategies of inertia weight in particle swarm optimisation. Six versions of particle swarm optimisation were analysed on 28 benchmark functions, prepared for the Special Session on Real-Parameter Single Objective Optimisation at CEC2013. The random components of tested inertia weight were generated from Beta distribution with different values of shape parameters. The best analysed PSO version is the multiswarm PSO, which combines two strategies of updating the inertia weight. The first is driven by the temporally varying shape parameters, while the second is based on random control of shape parameters of Beta distribution.

1. Introduction

The particle swarm optimisation—PSO—is a popular heuristic optimisation algorithm developed by Kennedy and Eberhart [1]. It is a nature inspired heuristic, which mimics the behaviour of flocks of birds or schools of fish. The recent survey of variants of PSO can be found in [2]. It is a population based evolutionary technique [3, 4], its introductory description is provided in [5]. The PSO has been successfully applied to many real life optimisation problems [6, 7].

Recently the PSO oriented research focuses on the development of new adaptation strategies, which avoid the premature convergence of particle population, or being trapped in local optima. For example the periodic changes of number of particles in population enhance the PSO performance [8]. The adaptive tuning of velocity particle estimated by the average velocity information accelerates the PSO ability to jump out of local optima [9]. Hu et al. [10] developed adaptive variant of PSO called PSO-MAM, which adopts the subgradient method for adjusting the PSO parameters. Liu et al. [11] applied in CPSO—chaotic particle swarm optimisation—the logistic equation for adjusting the new location of particles.

The improvement the estimation of particle's velocity is an essential task in PSO research. It was shown that the

inertia weight—IW—helps to increase the overall PSO search performance [4, 12]. Nickabadi et al. [13] provide the overview of 15 different strategies for the inertia weights adaptation.

The random adaptations of inertia weight play an important role in improving the PSO performance [4, 11, 12, 14, 15]. Mostly they support the exploratory search in the beginning of optimisation process. They increase the population diversity during the search process. Bansal et al. [14] compared 15 different IW strategies on 5 optimisation problems. The linear decreasing inertia term with logistic mapping was the best IW strategy in terms of average error. The logistic mapping of form $x[i + 1] = 4x[i](1 - x[i])$ is random number generator related to the symmetric Beta distribution with parameters $a = 0.5$ and $b = 0.5$ [16].

Besides the adaptation strategies of PSO parameters the special attention has to be put on development the multiswarm PSO [17–19]. The multiswarm PSO based on exclusion and anticonvergence was tested in dynamic environments [20]. The master slave multiswarm models with competitive and collaborative versions, in which the slave swarm provides the master swarm with the best particle, were studied in [17]. The cooperative multiswarm PSO of four swarms with cooperative search and diversity strategy performed better than single PSO on complex multimodal functions [21]. The five swarms with constant period of migration and

constant migration rate outperformed single PSO on eight optimisation problems [18].

The comparison study of 12 different migration strategies 6 on 36 optimisation problems is provided in [19]. Two migration strategies BW and BWM—the BWM applied the mutation on migrating particles—based on migrating the selected number of best particles from subswarm and substituting with them the worst particles outperformed remaining migration models. The parallel PSO with three communication strategies is compared in work of Chang et al. [22]. All three migration strategies are applied sequentially in one optimisation run and periodically exchange the particles between subswarms.

The aim of the presented paper is to compare selected version of PSO. The tested single and multiswarm versions of particle swarm optimisation are based on modifications of inertia weight, which are related to the random component controlled by the Beta distribution.

The remaining part of paper is arranged as follows. The description of PSO provides details on standard PSO, the proposed random inertia weight strategies, and the description of tested multiswarm PSO. Results comment on the finding based on extensive 10 dimensional computational experiments. The article summarizes the main findings in Conclusions.

2. The Description of PSO

2.1. The Standard PSO. The standard PSO (sPSO) modifies the location of particle $X^j = \{x_1^j, \dots, x_{\dim}^j\}$ with dimension \dim using the velocity $V^j = \{v_1^j, \dots, v_{\dim}^j\}$ updated in generation t as

$$\begin{aligned} v_i^j[t] &= w[t] v_i^j[t-1] + c_1 r_1 (g_i - x_i^j[t-1]) \\ &\quad + c_2 r_2 (p_i^j[t-1] - x_i^j[t-1]) \end{aligned} \quad (1)$$

for $i = 1, \dots, \dim$,

where the $r_1 \sim U(0, 1)$ and $r_2 \sim U(0, 1)$ are random numbers with uniform distribution, c_1, c_2 denote the acceleration coefficients of social and cognitive learning, and the $w[t]$ is the inertia weight.

The new location of particle is computed as

$$x_i^j[t] = x_i^j[t-1] + v_i^j[t] \quad \text{for } i = 1, \dots, \dim, \quad (2)$$

and the social component is controlled by the location of the global best particle denoted as $G = \{g_1, \dots, g_{\dim}\}$. For solving the minimization problem based on fitness function $f(X)$ is the $f(G) \leq f(X^j)$ for all X^j in population. The cognitive learning component is represented by the personal best location of particle $P^j = \{p_1^j, \dots, p_{\dim}^j\}$, which is $f(P^j) \leq f(X^j)$ for all actually known locations of particle $X^j[t]$. Equations (1) and (2) are applied on all X^j for $j = 1, \dots, n_{\text{pop}}$ with n_{pop} is the number of particles in swarm population [1, 3–5].

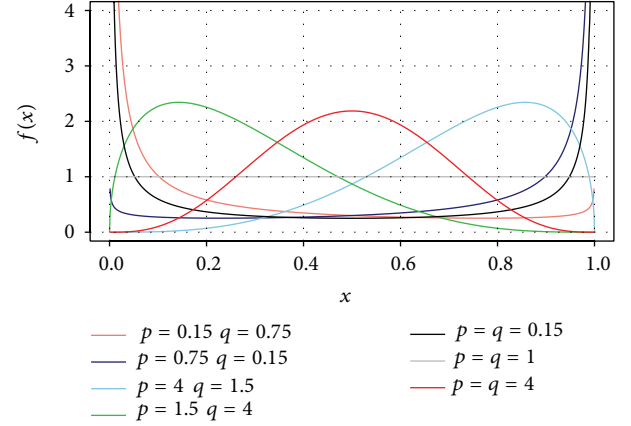


FIGURE 1: The selected densities of Beta distribution.

The sPSO is based on the velocity update with the linear decreasing inertia term $w[t]$, calculated with the formula

$$w[t] = (w_{\max} - w_{\min}) \frac{t_{\max} - t}{t_{\max}} + w_{\min}, \quad (3)$$

where the w_{\max} was set to 0.9, w_{\min} is equal to 0.4, and t_{\max} is the maximum number of generations.

The velocity update formula is restricted by v_{\max} , and it is applied as velocity control on the cases, when $|v_i^j| > v_{\max}$. Then the value of velocity is bounded on

$$v_i^j = \begin{cases} v_{\max} & \text{for } v_i^j > v_{\max} \\ -v_{\max} & \text{for } v_i^j < -v_{\max} \end{cases} \quad (4)$$

Note that this type of velocity control only enables limiting the maximum distance in which particle may move during one iteration [1, 23, 24].

2.2. The Proposed Inertia Weight Modifications. The proposed inertia weight modifications are based on random numbers generated using the Beta distribution. The density of Beta distribution $f(x)$ is defined as

$$f(x) = \begin{cases} \frac{1}{B(p, q)} x^{p-1} (1-x)^{q-1} & \text{for } 0 \leq x \leq 1 \\ 0 & \text{for } (0 > x) \wedge (x > 1). \end{cases} \quad (5)$$

Figure 1 shows selected densities of Beta distribution with different values of shape parameters p and q . The Beta distribution allows simulation from symmetric densities ($p = q$) and asymmetric densities with shape parameters $p \neq q$. Note that the uniform distribution is a special case of Beta distribution $p = q = 1$, and it has the maximum entropy from all Beta distributions.

One of the main advantages of Beta distribution is that it describes probability densities with various shapes on the interval $(0, 1)$. For equal shape parameters $p = q > 1$ the density is bell shaped, for $p = q > 1$ is U shaped. The U shaped densities allow simulating the extremes on interval

TABLE 1: Tested inertia weight updates based on Beta distribution.

PSO version	The weight update formula	Random component
RBld	$w[t] = \left((w_{\max} - w_{\min}) \frac{t_{\max} - t}{t_{\max}} + w_{\min} \right) - \frac{rb}{2}$	$rb \sim \text{Beta}(bs[t], bs[t])$
RBrr	$w[t] = \left((w_{\max} - w_{\min}) \frac{t_{\max} - t}{t_{\max}} + w_{\min} \right) - \frac{rb}{2}$	$rb \sim \text{Beta}(r_1, r_2)$
RBRa	$w[t] = \left((w_{\max} - w_{\min}) \frac{t_{\max} - t}{t_{\max}} + w_{\min} \right) + rb \cdot w_{\min}$	$rb \sim \text{Beta}(r_1, r_2)$

$(0, 1)$, while the bell shaped ones are focused on center of interval. This property supports the balanced exploratory and exploitative search process and avoids the premature convergence.

Table 1 shows definitions of three tested inertia weight strategies based on the Beta distribution. The RBld represents the linearly decreasing inertia weight with random component based on symmetric Beta distribution with linearly varied shape parameters $bs[t]$. The $bs[t]$ are controlled by the iteration t and are expressed as

$$bs[t] = (bs_{\max} - bs_{\min}) \frac{t - 1}{t_{\max} - 1} + bs_{\min}, \quad (6)$$

where the $bs[t] = p = q$ represents the shape parameters for symmetrical Beta distribution, which are applied on random number generation in time t .

The RBrr inertia weight version applies randomly selected shape parameters $r_1 \sim U(0, 1)$ and $r_2 \sim U(0, 1)$. The simulated random component for one generation consists mainly of random numbers generated from different asymmetrical Beta distributions. Note that the probability that $r_1 = r_2$ is smaller than the probability that $r_1 \neq r_2$.

The RBRa is modification of original of logistic mapping [11, 12]. The noise generated by the Beta distribution random generator is added to linearly varied inertia weight. The randomly varied shape parameters enable generation from both symmetrical and asymmetrical Beta distributions.

2.3. The Multiswarm PSO. The new proposed multiswarm PSO combines the search of four subswarms. This PSO version is marked as BrBl. The algorithm follows the principles of multiswarm algorithms [17–19], and it is completed by migration principle. The subswarms are divided into the two groups: the cooperative subswarms and elitistic subswarm. The subswarms use different inertia weight Beta distribution strategies. They share the information about global best particle only through the migration process.

The migration period is controlled by the simple rule, which increases the number of generations between two successive migrations. The migration iteration $t_m[i]$ is controlled by the previous migration $t_m[i - 1]$ and is calculated as follows:

$$t_m[i] = 2t_m[i - 1]. \quad (7)$$

This mechanism supports in the beginning of search process the exploration of search space through the intensive

migration of particles. The increase of $t_m[t]$ supports the exploitive search. The migration of cooperative and of elitistic subswarms is performed in the same generation.

The cooperative subswarms are formed of the three subswarms. Their cooperation is based on migration with migration rate m_r . Each subswarm selects n_m the number of its best particles in generation T_m and replaces the n_m randomly selected particles of swarm. The two cooperative swarms use the RBrr inertia weight update; the third cooperative swarm applies the RBld updating formula. The elitistic swarm uses the RBld inertia weight control.

The selection of subswarm for emigration is controlled randomly. Note that with the probability $p = 1/6$ all three subswarms will substitute their own worst or randomly selected particles with their own best particles, with probability $p = 2/3$ at least one of subswarm interchanges its worst or random particles with its bests, and with probability $p = 1/3$ the subswarm obtains best particles from other cooperative subswarms.

The second group of subswarms is formed from one elitistic swarm. This subswarm searches over the search space and receives the all best particles from cooperative swarms. The best particles substitute the randomly selected particles from elitistic subswarm. The elitistic swarm does not share the knowledge of global best particle with cooperative subswarms.

3. Results

The proposed modifications of inertia weight strategies were applied on 28 CEC2013 benchmark minimization problems [25]. Only 10 dimensional problems were analysed in the presented study. The set of CEC2013 benchmark problem consists of five unimodal functions f1–f5, fifteen multimodal problems f6–f20, and eight composition functions f21–f28.

The search space for all CEC2013 benchmark functions was $(-100, 100)$. Each PSO run was repeated 51 times per one optimisation problem. The maximum number of function evaluations was 100000, as recommended by the CEC2013 benchmark optimisation experiment [25].

The computations were made using the R statistical environment 3.0.2 [26] on 64-bit GNU/Linux operative system, and benchmark functions were used through the implementation of CEC2013 R package v0.1-4 [27]. The R package serves as a wrapper of original C code of 28 benchmark functions [25]. The random number generator was based on the work of Matsumoto and Nishimura [28].

TABLE 2: The minimum values achieved at 10E + 05 iteration and Min. is the problem solution.

CEC problem	Min.	RBRa	RBrr	AMPSO2	RBld	sPSO	BrBl
f1	-1400	-1400.00	-1400.00	-1400.00	-1400.00	-1400.00	-1400.00
f2	-1300	31802.55	10719.58	204814.76	20573.61	51187.06	18609.16
f3	-1200	-1199.74	-1199.95	17054.84	-1199.82	-1199.96	-1199.95
f4	-1100	-964.04	-947.26	576.38	-981.92	-913.96	-388.49
f5	-1000	-1000.00	-1000.00	-1000.00	-1000.00	-1000.00	-1000.00
f6	-900	-899.93	-899.99	-899.51	-899.81	-899.76	-900.00
f7	-800	-799.96	-799.32	-798.57	-799.92	-799.95	-798.90
f8	-700	-679.83	-679.80	-679.79	-679.78	-679.84	-679.94
f9	-600	-599.25	-599.23	-599.25	-598.43	-599.43	-599.37
f10	-500	-499.93	-499.95	-499.82	-499.91	-499.91	-499.94
f11	-400	-400.00	-400.00	-397.56	-399.01	-400.00	-400.00
f12	-300	-295.03	-294.03	-284.82	-293.04	-295.03	-295.03
f13	-200	-193.73	-193.73	-182.54	-193.04	-197.16	-192.22
f14	-100	-96.40	-96.40	-40.45	-93.11	-93.05	-93.12
f15	100	243.94	382.90	624.16	343.59	463.56	295.58
f16	200	200.70	200.24	200.38	200.47	200.58	200.10
f17	300	303.29	301.60	324.08	302.45	304.34	301.11
f18	400	414.78	413.57	433.38	414.77	415.52	414.11
f19	500	500.04	500.07	500.89	500.19	500.31	500.14
f20	600	601.96	601.94	602.44	602.03	601.81	602.11
f21	700	800.00	800.00	800.03	800.00	700.00	800.00
f22	800	829.38	823.36	925.66	829.06	830.92	821.09
f23	900	1149.35	1080.55	1613.80	1127.72	1181.06	1338.13
f24	1000	1109.13	1114.66	1125.87	1112.24	1136.31	1109.72
f25	1100	1301.93	1303.37	1302.83	1302.10	1301.53	1213.63
f26	1200	1307.96	1305.97	1321.06	1306.97	1308.95	1304.98
f27	1300	1615.16	1622.81	1658.37	1636.13	1607.02	1603.81
f28	1400	1500.00	1500.00	1500.01	1500.00	1500.00	1500.00

The single PSO parameter settings were based on [3, 24]. The size of population was $n_{\text{pop}} = 40$, the $c_1 = c_2 = 2$, and $v_{\text{max}} = 95$. The populations were randomly initialized within the search space using the uniform distribution and the values of parameters controlling inertia weight were $w_{\text{max}} = 0.9$ and $w_{\text{min}} = 0.4$. The linearly increasing values of shape parameters were $bs_{\text{min}} = 0.1$, $bs_{\text{max}} = 4$, and $t_{\text{max}} = 100000$.

The PSO with proposed inertia weight strategies was compared with standard PSO (sPSO) and AMPSO2. The AMPSO2 uses the Beta distribution on adaptive mutation of the personal best particles and global best particle [29]. The RBrr and RBRa use the shape parameters r_1 and r_2 randomly generated from interval (0, 1).

The parameter settings particle initializations of BrBl subswarms were those used in single PSO. The BrBl migration rate $m_r = 0.2$, the first migration started in the second generation $t_m[1] = 2$, and the total number of realized migrations was 9.

3.1. The Exploration and Exploitation of Proposed PSO Versions. We relate the description of balance between the exploration and exploitation to the evolution of the variances and fitness values of global best particles generated by the all 51 optimisation runs.

The variance of tested PSO versions was described using the standard deviations of differences between fitness values and median, which was obtained from 51 runs in given iteration. The results for the first 4000 iterations on 12 selected benchmark problems are shown in Figures 2, 3, and 4.

On unimodal problems f1–f5 and multimodal problem f17 the AMPSO2 and sPSO show clearly different patterns in the evolution of standard deviations than PSO versions with Beta distribution. The PSO versions with Beta distribution show the decrease of the variance of swarm particles, while the AMPSO2 and sPSO show the stagnation. These similar patterns of decrease and stagnation are apparent on the fitness values of global best particles.

Those patterns are connected to the convergence of tested PSO versions. For example on f1 problem all PSO versions based on Beta distribution found earlier the optimum than AMPSO2 and sPSO (see the results of Table 3).

The BrBl shows the highest variances in the beginning of iteration search. These are connected to the intensive migrations, performed during the early stages of optimisation search. The main benefit is shown in later rapid decrease of fitness value (e.g., see the results in Figure 2). Similar patterns are shown in [17, 21, 30].

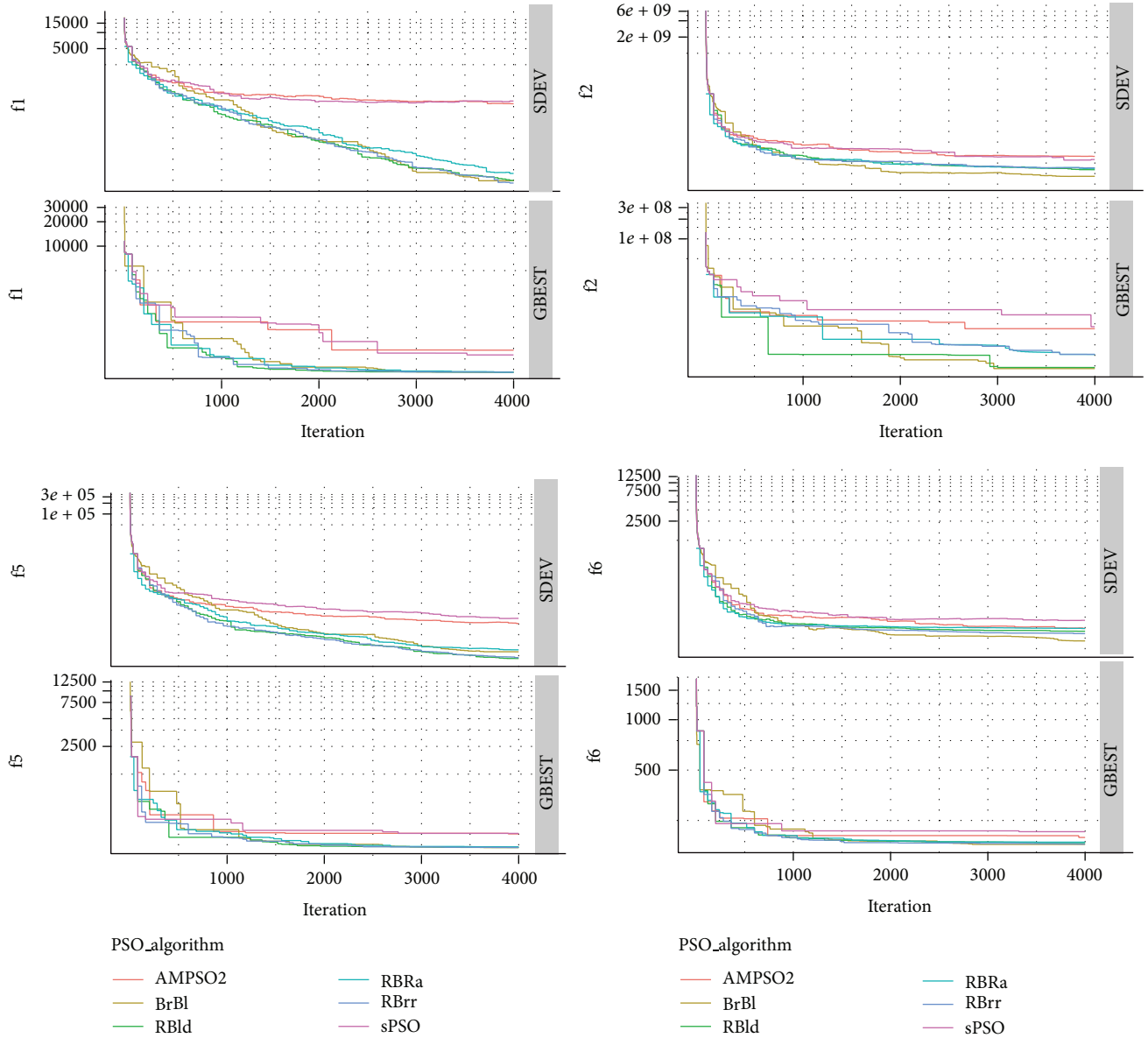


FIGURE 2: The normalized standard deviation and global best model of 51 runs for f1, f2, f5, and f6; SDEV is the standard deviation, GBEST is the standard deviation, GBEST fitness of global best particle. Note: all values are shifted due to the logarithmic transformation of y -axis.

The BrBl version also shows the increase of variance during the search process on f15 and f23. This fact is again connected to the finding of better solutions in terms of values of global best particle (see Table 3). On the other hand the BrBl was on f23 the second worst PSO version (see Table 2).

3.2. The Comparison of PSO Versions. Following the recommendation of CEC2013 the maximum function evaluation (FES) was set as $10E + 05$ [24, 25, 31]. The overall results of fitness values of global best particles are shown Table 3 for FES = $10E + 04$ and Table 2 for FES = $10E + 05$.

The PSO versions with Beta distribution components show the best convergence properties on all benchmark problems for FES $10E + 04$ (see Table 3). The best fitness

values recorded on FES $10E + 04$ show the RBld for unimodal problems. Three PSO algorithms achieved best fitness values on 15 multimodal problems, f6–f21. They are sPSO on f8, RBld on f13 and f21, and RBrr on f7, f10, f12, and f20. The remaining 8 multimodal problems were described by the BrBl. The RBld and BrBl were superior for composition functions, f21–f28.

The results of FES $10E + 05$ show that tested versions of PSO solved the following benchmark problems: f1—all PSO versions, f5—all PSO versions, f6—the BrBl version, f11—all PSO versions except the AMPSO2, and RBld, f21—sPSO. These results are comparable with findings of Zambrano-Bigiarini et al. [24] and El-Abd [31]. The BrBl achieved the best values on 13 optimisation problems.

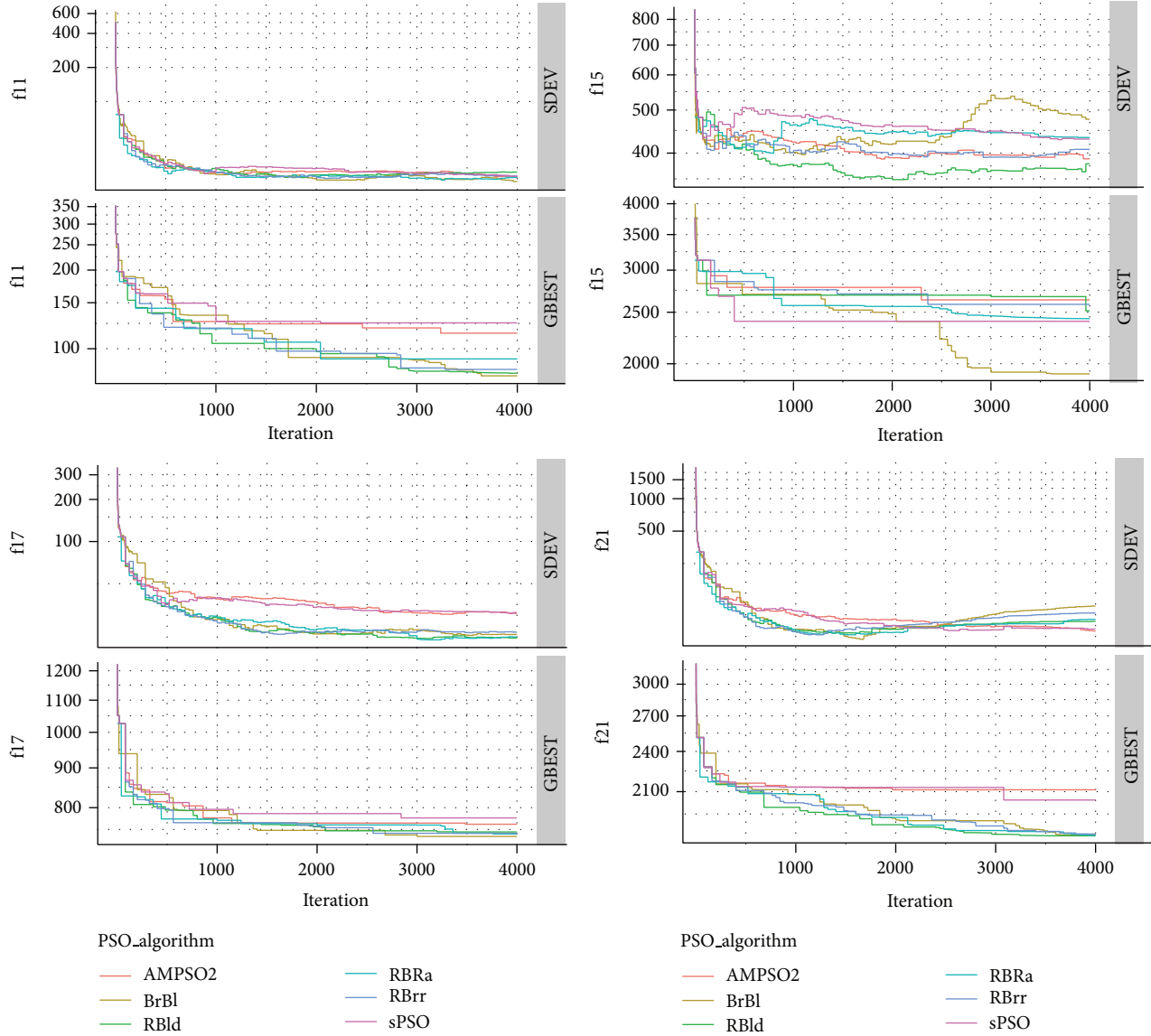


FIGURE 3: The normalized standard deviation and global best model of 51 runs for f11, f15, f17, and f21; SDEV is the standard deviation, GBEST fitness of global best particle. Note: all values are shifted due to the logarithmic transformation of y-axis.

The comparison of mean performance of all 51 runs for FES $10E + 04$ is shown in Table 4. Those results are based on the contrast test of unadjusted median test (for detailed explanation see [32]). The final ranking shows that the BrBl PSO version is superior to the remaining tested versions. Similar results showed the contrast test values obtained for FES $10E + 05$.

These finding were confirmed by the results of paired Wilcoxon test. The P values for Wilcoxon test of BrBl and other PSO versions were statistically significant for FES $10E + 04$ (see Table 5) and also for FES $10E + 05$.

4. Conclusions

The presented analysis evaluates the 6 different versions of PSO algorithm on 28 CEC2013 benchmark functions. The goal was to experimentally compare the different inertia

weight updating strategies related to the random component generated by the Beta distribution.

The computational experiment consists of approximately $8.57E + 08$ function evaluations (28 benchmark functions \times 51 repetitions \times 100000 function evaluations \times 6 versions of particle swarm optimization algorithm). We compared 5 single swarm PSO versions and 1 multiswarm PSO version.

The results of comparison of selected single swarm PSO versions indicate that the Beta distribution applied on inertia weight strategy provides important source of modifications of original PSO. It supports the balanced exploratory and exploitive search. The best single swarm strategies according to the results of contrast test based on unadjusted median are RBld and RBrr.

Our results highlight that the best version from 6 tested PSO modifications is the multiswarm algorithm BrBl. The BrBl combines the swarms with modifications of inertia

TABLE 3: The minimum values achieved at $10E + 04$ iteration and Min. is the problem solution.

CEC problem	Min.	RBRa	RBrr	AMPSO2	RBld	sPSO	BrBl
f1	-1400	-1400.00	-1400.00	-1147.93	-1400.00	-1265.72	-1400.00
f2	-1300	222040.73	322300.27	1498763.45	130380.94	3959345.31	236444.51
f3	-1200	2136685.60	392088.96	1193329345.61	369651.03	331615655.13	6232942.40
f4	-1100	5620.73	5057.65	6101.20	4597.78	7088.07	5451.37
f5	-1000	-999.98	-1000.00	-943.64	-1000.00	-961.61	-999.97
f6	-900	-899.70	-899.84	-878.23	-899.57	-881.82	-899.94
f7	-800	-795.95	-797.12	-752.51	-794.56	-771.83	-788.15
f8	-700	-679.73	-679.75	-679.75	-679.78	-679.84	-679.73
f9	-600	-598.53	-597.77	-593.72	-597.54	-593.42	-599.29
f10	-500	-498.94	-499.44	-464.60	-499.36	-468.55	-498.81
f11	-400	-394.23	-395.61	-368.60	-395.74	-362.84	-395.99
f12	-300	-283.02	-292.15	-252.46	-292.11	-258.93	-286.85
f13	-200	-182.06	-183.29	-147.10	-188.58	-148.35	-179.85
f14	-100	-43.56	116.03	199.00	73.64	1221.41	25.31
f15	100	1366.48	859.64	1416.16	834.71	1331.53	505.67
f16	200	201.00	200.58	200.73	200.87	200.89	200.40
f17	300	327.16	322.55	366.99	323.63	365.88	318.69
f18	400	446.89	438.03	473.89	435.77	482.47	433.20
f19	500	501.16	501.02	508.15	500.75	505.62	500.55
f20	600	602.94	602.78	603.07	602.86	603.40	602.98
f21	700	803.41	800.72	1080.72	800.49	1053.29	801.40
f22	800	1107.86	1009.20	1318.77	870.95	2022.70	1003.78
f23	900	1580.41	1796.86	2286.08	1740.24	2546.45	1514.33
f24	1000	1137.37	1144.17	1175.66	1130.14	1172.10	1135.56
f25	1100	1305.98	1305.48	1315.70	1303.40	1319.22	1257.89
f26	1200	1321.41	1321.35	1347.29	1311.11	1357.60	1308.38
f27	1300	1646.86	1641.18	1799.34	1659.35	1776.92	1637.37
f28	1400	1503.26	1500.31	1692.99	1500.23	1667.17	1500.63

TABLE 4: The contrast test on best values achieved on $10E + 04$ iteration.

	RBRa	RBrr	AMPSO2	RBld	sPSO	BrBl	Ranking
RBRa	—	$2.67E + 00$	$-4.14E + 01$	$2.56E + 00$	$-4.74E + 01$	$7.66E + 00$	4
RBrr	$-2.67E + 00$	—	$-4.41E + 01$	$-1.09E - 01$	$-5.01E + 01$	$4.99E + 00$	2
AMPSO2	$4.14E + 01$	$4.41E + 01$	—	$4.40E + 01$	$-6.00E + 00$	$4.91E + 01$	5
RBld	$-2.56E + 00$	$1.09E - 01$	$-4.40E + 01$	—	$-5.00E + 01$	$5.10E + 00$	3
sPSO	$4.74E + 01$	$5.01E + 01$	$6.00E + 00$	$5.00E + 01$	—	$5.51E + 01$	6
BrBl	$-7.66E + 00$	$-4.99E + 00$	$-4.91E + 01$	$-5.10E + 00$	$-5.51E + 01$	—	1

TABLE 5: The Wilcoxon test on best values achieved on $10E + 04$ iteration.

	RBRa	RBrr	AMPSO2	RBld	sPSO	BrBl	Ranking
RBRa	—	$1.00E + 00$	$1.51E - 110$	$1.00E + 00$	$1.90E - 176$	$1.00E + 00$	4
RBrr	$4.38E - 18$	—	$7.75E - 141$	$4.57E - 01$	$2.22E - 184$	$1.00E + 00$	2-3
AMPSO2	$1.00E + 00$	$1.00E + 00$	—	$1.00E + 00$	$1.38E - 09$	$1.00E + 00$	5
RBld	$3.02E - 18$	$5.43E - 01$	$7.89E - 140$	—	$2.79E - 181$	$1.00E + 00$	2-3
sPSO	$1.00E + 00$	$1.00E + 00$	$1.00E + 00$	$1.00E + 00$	—	$1.00E + 00$	6
BrBl	$5.57E - 48$	$6.00E - 13$	$5.01E - 201$	$4.60E - 18$	$6.51E - 211$	—	1

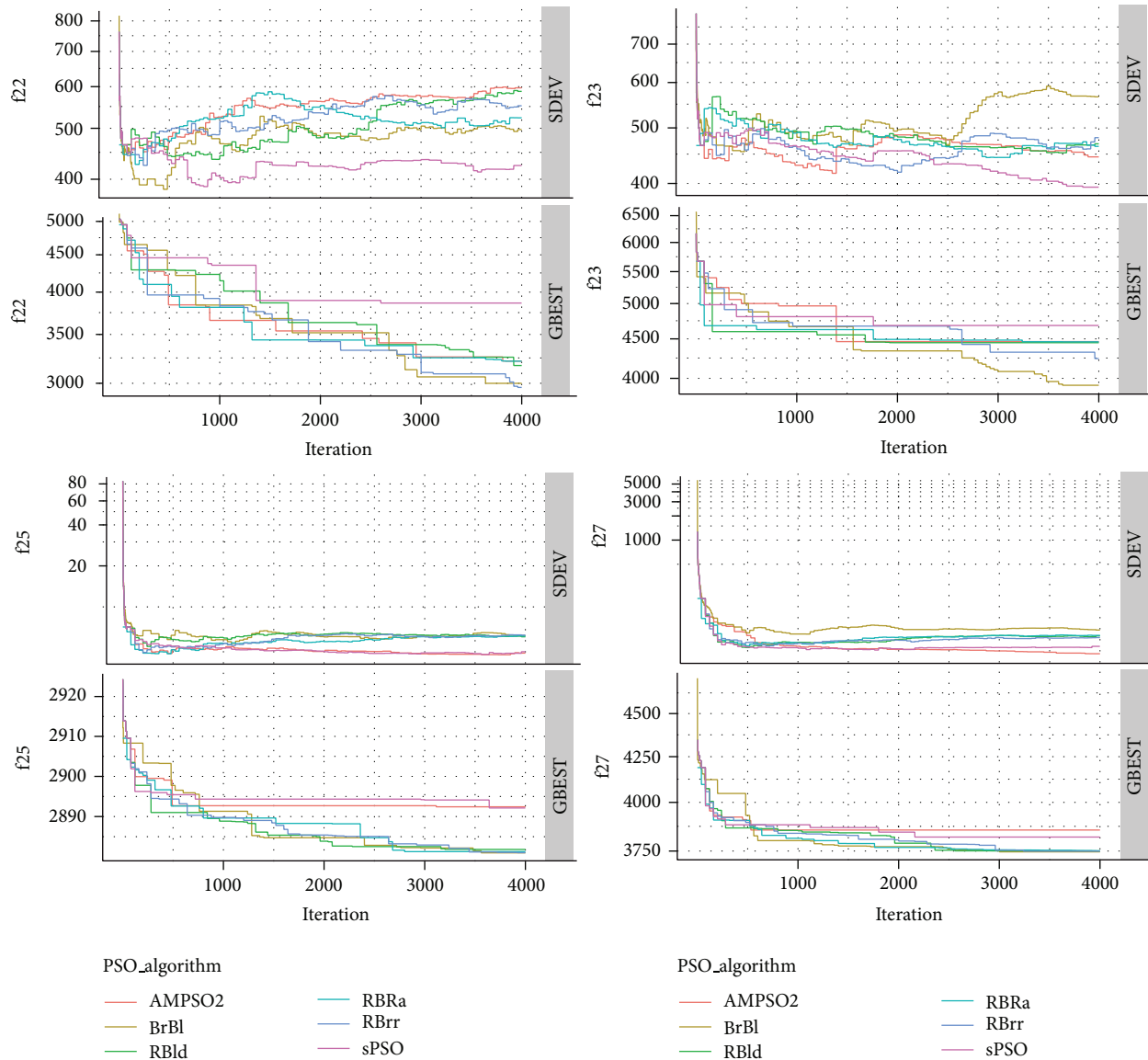


FIGURE 4: The normalized standard deviation and global best model of 51 runs for f22, f23, f25, and f27; SDEV is the standard deviation, GBEST fitness of global best particle. Note: all values are shifted due to the logarithmic transformation of y-axis.

weight by the random component controlled by the time varied constant shape parameters and randomly varied shape parameters of Beta distributions.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, The University of Western Australia, Perth, Australia, December 1995.
- [2] K. Kameyama, "Particle swarm optimization—a survey," *IEICE Transactions on Information and Systems*, vol. 92, no. 7, pp. 1354–1361, 2009.
- [3] Y. Shi and R. Eberhart, "Modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '98)*, pp. 69–73, IEEE Computer Society, Washington, DC, USA, May 1998.
- [4] R. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms," in *Proceedings of the Congress on Evolutionary Computation*, vol. 1, pp. 94–100, 2001.
- [5] A. P. Engelbrecht, *Computational Intelligence: An Introduction*, John Wiley & Sons, New York, NY, USA, 2nd edition, 2007.
- [6] J.-B. Park, Y.-W. Jeong, J.-R. Shin, and K. Y. Lee, "An improved particle swarm optimization for nonconvex economic dispatch

- problems," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 156–166, 2010.
- [7] M. Zambrano-Bigiarini and R. Rojas, "A model-independent particle swarm optimisation software for model calibration," *Environmental Modelling & Software*, vol. 43, pp. 5–25, 2013.
 - [8] D. Chen and C. Zhao, "Particle swarm optimization with adaptive population size and its application," *Applied Soft Computing Journal*, vol. 9, no. 1, pp. 39–48, 2009.
 - [9] G. Xu, "An adaptive parameter tuning of particle swarm optimization algorithm," *Applied Mathematics and Computation*, vol. 219, no. 9, pp. 4560–4569, 2013.
 - [10] M. Hu, T. Wu, and J. D. Weir, "An adaptive particle swarm optimization with multiple adaptive methods," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 705–720, 2013.
 - [11] B. Liu, L. Wang, Y.-H. Jin, F. Tang, and D.-X. Huang, "Improved particle swarm optimization combined with chaos," *Chaos, Solitons and Fractals*, vol. 25, no. 5, pp. 1261–1271, 2005.
 - [12] Y. Feng, G.-F. Teng, A.-X. Wang, and Y.-M. Yao, "Chaotic inertia weight in particle swarm optimization," in *Proceedings of the 2nd International Conference on Innovative Computing, Information and Control (ICICIC '07)*, p. 475, Kumamoto, Japan, September 2007.
 - [13] A. Nickabadi, M. M. Ebadzadeh, and R. Safabakhsh, "A novel particle swarm optimization algorithm with adaptive inertia weight," *Applied Soft Computing Journal*, vol. 11, no. 4, pp. 3658–3670, 2011.
 - [14] J. C. Bansal, P. K. Singh, M. Saraswat, A. Verma, S. S. Jadon, and A. Abraham, "Inertia weight strategies in particle swarm optimization," in *Proceedings of the 3rd World Congress on Nature and Biologically Inspired Computing (NaBIC '11)*, pp. 633–640, October 2011.
 - [15] M. Jakubcová, P. Máca, and P. Pech, "A comparison of selected modifications of the particle swarm optimization algorithm," *Journal of Applied Mathematics*, vol. 2014, Article ID 293087, 10 pages, 2014.
 - [16] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 261, no. 5560, pp. 459–467, 1976.
 - [17] B. Niu, Y. Zhu, X. He, and H. Wu, "MCP SO: a multi-swarm cooperative particle swarm optimizer," *Applied Mathematics and Computation*, vol. 185, no. 2, pp. 1050–1062, 2007.
 - [18] M. Gang, Z. Wei, and X. Chang, "A novel particle swarm optimization algorithm based on particle migration," *Applied Mathematics and Computation*, vol. 218, no. 11, pp. 6620–6626, 2012.
 - [19] X. Lai and G. Tan, "Studies on migration strategies of multiple population parallel particle swarm optimization," in *Proceedings of the 8th International Conference on Natural Computation (ICNC '12)*, pp. 798–802, IEEE, May 2012.
 - [20] T. Blackwell and J. Branke, "Multiswarms, exclusion, and anti-convergence in dynamic environments," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 459–472, 2006.
 - [21] J. Zhang and X. Ding, "A multi-swarm self-adaptive and cooperative particle swarm optimization," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 6, pp. 958–967, 2011.
 - [22] J.-F. Chang, S.-C. Chu, J. F. Roddick, and J.-S. Pan, "A parallel particle swarm optimization algorithm with communication strategies," *Journal of Information Science and Engineering*, vol. 21, no. 4, pp. 809–818, 2005.
 - [23] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.
 - [24] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas, "Standard particle swarm optimisation 2011 at CEC-2013: a baseline for future PSO improvements," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '13)*, pp. 2337–2344, Cancun, Mexico, June 2013.
 - [25] J. J. Liang, B. Y. Qu, P. N. Suganthan, and A. G. Hernandez-Diaz, "Problem definitions and evaluation criteria for the CEC 2013 special session and competition on real-parameter optimization," Tech. Rep. 201212, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, 2013.
 - [26] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2013.
 - [27] M. Zambrano-Bigiarini and Y. Gonzalez Fernandez, "cec2013: benchmark functions for the special session and competition on real-parameter single objective optimization at CEC-2013," R Package Version 0.1-4, 2013.
 - [28] M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Transactions on Modeling and Computer Simulation*, vol. 8, no. 1, pp. 3–30, 1998.
 - [29] M. Pant, R. Thangaraj, and A. Abraham, "Particle Swarm Optimization using adaptive mutation," in *Proceedings of the 19th International Conference on Database and Expert Systems Applications (DEXA '08)*, pp. 519–523, IEEE Computer Society, September 2008.
 - [30] H. Wang, H. Sun, C. Li, S. Rahnamayan, and J. Pan, "Diversity enhanced particle swarm optimization with neighborhood search," *Information Sciences*, vol. 223, pp. 119–135, 2013.
 - [31] M. El-Abd, "Testing a particle swarm optimization and artificial bee colony hybrid algorithm on the CEC13 benchmarks," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '13)*, pp. 2215–2220, IEEE Computational Intelligence Society, Cancun, Mexico, June 2013.
 - [32] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

