

## PAPER

# Enhanced Particle Swarm Optimization with Self-Adaptation on Entropy-Based Inertia Weight

Hei-Chia WANG<sup>†a)</sup>, Nonmember and Che-Tsung YANG<sup>†</sup>, Student Member

**SUMMARY** The inertia weight is the control parameter that tunes the balance between the exploration and exploitation movements in particle swarm optimization searches. Since the introduction of inertia weight, various strategies have been proposed for determining the appropriate inertia weight value. This paper presents a brief review of the various types of inertia weight strategies which are classified and discussed in four categories: static, time varying, dynamic, and adaptive. Furthermore, a novel entropy-based gain regulator (EGR) is proposed to detect the evolutionary state of particle swarm optimization in terms of the distances from particles to the current global best. And then apply proper inertia weights with respect to the corresponding distinct states. Experimental results on five widely applied benchmark functions show that the EGR produced significant improvements of particle swarm optimization.

**key words:** adaptive inertia weight, exploration-exploitation trade-off, particle swarm optimization, entropy

## 1. Introduction

Particle swarm optimization (PSO) is a relatively new heuristic algorithm proposed in 1995 [1]. In the context of PSO, a swarm consists of a group of randomly initialized individuals, called particles, that represents a population of potential solutions to a given optimization problem. Each randomly initialized particle moves toward the optimal location with an iteratively updated velocity in the search space. The velocity of a particle is calculated using Eq. (1):

$$V_i(t+1) = V_i(t) + c_1 r_1 (pBest - X_i(t)) + c_2 r_2 (gBest - X_i(t)) \quad (1)$$

where  $V_i(t)$  represents the current velocity,  $V_i(t+1)$  is the new velocity,  $c_1$  and  $c_2$  are the acceleration coefficients that weight the cognitive and social components.  $r_1$  and  $r_2$  are independent random numbers in the range [0, 1].

The new velocity consists of three components, including the previous velocity and two attracting forces. The so-called cognitive learning attracts a particle towards the individual known-best. Whereas, the social learning attracts a particle towards the global known-best, which has been experienced and shared in community. Iteratively, each randomly initialized particle is driven by both cognitive and social components towards the global known-best in the search space.

The position of the particle and the memory of the particle's known-best are updated using Eqs. (2) and (3).

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (2)$$

$$pBest_i(t+1) = \begin{cases} f(x_i(t+1)), & f(x_i(t+1)) > pBest_i(t) \\ pBest_i(t), & f(x_i(t+1)) \leq pBest_i(t) \end{cases} \quad (3)$$

where  $X_i(t)$  is the current position of the particle,  $pBest_i(t+1)$  is the personal best position of the particle achieved so far, and  $pBest_i(t)$  is the new best position.

The global known-best position is updated using Eq. (4).

$$gBest(t+1) = \text{Max}(pBest_i(t+1)) \quad (4)$$

where  $i$  is the index of particles ranging from 0 to  $n$ . The  $pBest$  and  $gBest$  are updated in iterations to keep the known-best in the memory of the individual and the community. These processes repeat iteratively until either the optimal location within a predefined threshold is achieved or the limit on the number of iterations is reached.

In the original PSO, the new velocity is derived from the combination of current velocity with the drives of cognitive learning and social interaction without inertia weight. The concept of inertia weight was first introduced in 1998 with a constant inertia weight [2], shown as Eq. (5).

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 (pBest - X_i(t)) + c_2 r_2 (gBest - X_i(t)) \quad (5)$$

The parameter  $\omega$ , inertia weight, was introduced to balance the exploration and exploitation of a particle in PSO. It is notable that a large inertia weight facilitates a global search, while a small inertia weight facilitates a local search [3], [4]. Exploration is the ability to conduct a search with diversity on a global scale. Exploitation is the ability to concentrate the search area around a promising candidate solution in order to locate the optimum precisely [5].

The fine-tuning balance between the local search and global search is crucial to the performance of heuristic algorithms [6]. However, most of studies apply the inertia weight ignoring the existing inhomogeneous of varying scatters of data samples in iterations. In this sense, a novel inertia weight strategy applying distinct inertia weight based on the measure expressing the evolutionary state in terms of the distances from particles to the expected goal is proposed to enhance the adaptability of particles during evolutions. Experimental results show that the novel inertia strategy outperforms the static and time-varying approaches.

Manuscript received August 4, 2015.

Manuscript revised October 17, 2015.

Manuscript publicized November 19, 2015.

<sup>†</sup>The authors are with Institute of Information Management, National Cheng Kung University, Taiwan.

a) E-mail: hcwang@mail.ncku.edu.tw

DOI: 10.1587/transinf.2015EDP7304

In this paper, various types of the previous works about inertia weight strategies are classified into four categories and reviewed in Sect.2. Section 3 describes the novel entropy-based inertia weight. Section 4 presents the experiments and results. Section 5 delivers the conclusions.

## 2. Review on Inertia Weight Strategies

Inertia weight plays a key role in facilitating the balance between exploration and exploitation searches in PSO. The larger inertia weight facilitates searches in global scale, while the smaller inertia weight tend to produce local searches in a smaller scale. It is recommended in heuristic optimization technique that the exploration is more suitable at the initial stages of searches and then shift to exploitation around promising solutions gradually as the search progresses [7]. In the early stages, exploration with great diversity can prevent becoming trapped by a local optimal. While during the later stages, exploitation in small scale is more suitable to reach the optimum solution efficiently for better convergence speed [8].

In this section, various types of PSO inertia weight strategies are reviewed briefly in four categories: static, time varying, dynamic, and adaptive which are shown as Table 1.

Static inertia weight is the first attempt and has been shown that able to improve performance of original PSO. It was reported that PSO with static inertia weight in the range [0.9, 1.2] can improve performance [2]. Nevertheless, a critical drawback of static inertia weight lies in problem dependency. Since, the association between inertia weight and problem optimality is usually ambiguous or difficult to discover, the appropriate inertia weight is therefore usually unknown [9].

Instead of using a fixed value of inertia weight, the time-varying strategies produce the value of inertia weight in a function of the iteration number. Time-varying strategies are the most majority of the variants of modified PSO [10] for introducing the more efficient inertia weight. The linearly decreasing strategy with values from 0.9 to 0.4 is recommended to produce excellent results for enhancing the performance of PSO efficiently [4].

In addition to linearly strategy, various types of non-linear decreasing strategies are reported combining the logarithmic, the exponential, and the simulated annealing techniques. Some exponential type of inertia weight is proposed in [11]–[13].

In a dynamic system, the objective function keeps changing and the optimum position is supposed to move over time. The traditional PSO has difficulty in tracking the dynamic goal [14]. To address this problem, a random inertia weight strategy [15] is proposed to enhance PSO for tracking the moving optima in a dynamic environment.

Another dynamic inertia weight strategy is the chaotic inertia weight approach. The concept of “chaos” stands for a non-linear system exhibiting highly dependence on initial conditions and compromising infinite unstable periodic motions within a bounded range. A general procedure

**Table 1** Types of PSO inertia weight strategies.

Category	Ref.	Formula
Static	[2]	$\omega \in [0.9, 1.2]$
Time-varying	[4]	$\omega = \omega_{min} - (\omega_{max} - \omega_{min}) \times \frac{g}{G}$
	[11]	$\omega = \omega_{min} + (\omega_{max} - \omega_{min}) \times e^{-\frac{10t}{iter_{max}}}$
	[11]	$\omega = \omega_{min} + (\omega_{max} - \omega_{min}) \times e^{-\frac{4t}{iter_{max}}}$
	[12]	$\omega = e^{-ak}$
	[13]	$\omega = \omega_0 \times e^{-a(\frac{g}{G})^b}$
Dynamic	[15]	$\omega = 0.5 + \frac{Rand()}{2}$
	[16]	$C \omega_k = \omega_k \times \gamma^k$ $(\omega_{min} - \omega_{max}) \frac{t}{iter_{max}} + \omega_{max} \times z$
	[17]	$\omega = 0.5 \times rand() + 0.5 \times z$ $z_n + I = 4z_n(1 - z_n)$ $z_n \in [0, 1]$ , where $n = 0, 1, 2, \dots$
	[18]	$\omega = \left[ 3 - \frac{1}{e^{\frac{S}{200}}} + \left( \frac{R}{8D} \right)^2 \right]^{-1}$ S is the swarm size, D is the dimension of the solution space, and R represents the fitness ranking of the given particle.
Adaptive	[19]	$\omega_i = (1.1 - \frac{gbest_i}{(pbest_i)_{average}})$ $C = (1 + \frac{gbest_i}{pbest_i})$ $V_i(t) = \omega V_i(t-1) + cr_i(pBest + gBest - 2X_i(t))$
	[20]	$\omega_i = \omega_{min} + \frac{(\omega_{max} - \omega_{min}) * Rank_i}{Total\ population}$
	[21]	$\omega = \begin{cases} \omega_{min} + \frac{(\omega_{max} - \omega_{min})(f - f_{min})}{f_{avg} - f_{min}}, & f \leq f_{avg} \\ \omega_{max}, & f > f_{avg} \end{cases}$ f is current object value

for chaotic search can be found in [16]. A chaotic inertia weight strategy using the benefits of better ability of mountain climbing and escape from a local optimum in the evolutionary process. In [17], the chaotic inertia weight strategy compromises the simple chaotic motions to the linear inertia weight and random inertia weight to produce the perturbed inertia weight.

In PSO, the evolutionary state would vary at different stages instead of merely varying with time. The common shared concept of adaptive inertia weight strategies is to detect the evolutionary state and then apply proper inertia weights with respect to the corresponding distinct states. The value of an adaptive inertia weight is usually determined on the measures that exhibiting the state of evolution or situation of searching.

A self-adaptive inertia weight [18] introduces a function relates the inertia weight to the size of the population dimension to compute the value of inertia weight adaptively.

The global-local best acceleration coefficient [19] computes the ratio of the global best fitness over the average local best fitness to determine the inertia weight in each iteration to prevent the problem of premature convergence. [20]

adjusts particle's position considering the particle's ranking in the swarm so that the most best particle moves slowly when compared to the least fitted particle. The adaptive inertia weight  $\omega$  proposed in [21] is varied depending on the particle's fitness value. The particles with good fitness tend to perform exploitation to refine the found results by local search. The particles with inferior fitness value tend to explore the solution space with large steps.

### 3. Entropy-Based Adaptive Inertia Weight

A novel adaptive inertia weight strategy involves using entropy-based gain (EG) is proposed. The EG aims to detect the evolutionary state of PSO in terms of the distances from particles to the current global best which is defined as:

$$EG_i = - \sum_j^n \left( \frac{d(x_j, gbest_i)}{\sum_{j=1}^n d(x_j, gbest_i)} \log \frac{d(x_j, gbest_i)}{\sum_{j=1}^n d(x_j, gbest_i)} \right) \quad (6)$$

Furthermore, the entropy-based gain regulator (EGR) weight which is defined as the reciprocal of EG, is used to adjust the inertia weight as:

$$EGR_i = \frac{1}{EG_i} \quad (7)$$

$EGR_i$  is the entropy-based gain regulator of the  $i$ -th iteration.  $d(x_j, gbest_i)$  is the distance between the  $j$ -th particle and the global best in the  $i$ -th iteration. The EGR is expected to produce turbulences with adequate amplitude at proper iterations to improve the performance of PSO. The EGR is designed to control the balancing of exploration and exploitation by adjusting the inertia weight adaptively.

The value of the EG will be greater while the distances from particles to the global best are equidistant (Fig. 1 left) than the distances are diverse (Fig. 1 middle). The smallest EG value is situations of the distances are in extremely great differences (Fig. 1 right).

In the left of Fig. 1, referred as the compact state, the EG value is significantly great and trends toward to conduct searches of exploitations with smaller ERG adjusted inertia weights since particles are all seem to be good and equally. In addition to that, giant movements of explorations are improper for convergence in this situation. In the middle of Fig. 1, referred as the diverse state, the EG value is relatively great and probably is the general initial situations of the PSO evolutions. Whereas the right of Fig. 1, referred as

the leap state, exhibits the scenario of the smallest EG occurred while a new global best is found currently. A new found far-moved global best decreases the EG value sharply and produce larger EGR and inertia weights with which are supposed to generates explorations. Therefore, the EGR controls the appropriate trade-off between exploration and exploitation depending upon the evolutionary state of the particles.

In general, the adaptive mechanism of inertia weight based on EG is considered as begin with diverse state in the early iterations and trends toward the compact state iteratively. Occasionally, the leap state cause turbulences resulting from the EG decreased sharply while a new far-moved global best occurred. And then followed by blending exploitations and explorations continuously toward the compact state until another new far-moved global best is happened occasionally or converge to the ever global best iteratively.

## 4. Experiments

### 4.1 Experimental Settings

The proposed EGR inertia weight (EGR-IW) strategy is validated with a number of analytical benchmark functions which have been extensively used in optimization problems [23]. The selected test functions, Ackley, Sphere, Rosenbrock, Rastrigin and Griewank, are considered as popular ones which are extendable to arbitrary dimensionality allowing for scaling investigations [24]. Each benchmark function was tested with a population of 30 particles and  $c1 = c2 = 2$  for a maximum of 1000 iterations as the stopping criteria. Fifty independent trials of each test function with respect to distinct inertia weight strategies on different test functions were executed. All of the employed benchmark functions and corresponding experimental settings are listed in Table 2.

Two strategies, a linear decreasing inertia weight (LDIW) varying from 0.9 to 0.4 and a random inertia weight (RIW) are used in comparisons to the novel inertia weight strategy with EGR. The comparative criteria of experimental results are presented in terms of the precision, success rate, and convergent period. The precision is in terms of the statistical results of the obtained optimum value in each trial. The success rate is the ratio of 50 independent trials that achieved the desired goals within a tolerance of the given threshold. The convergent periods express in terms of the number of iterations that were required to reach the goals in a limited predefined number of iterations. The projects are conducted in predefined limited iterations,  $G_{max} = 1000$ , maximum generations with different dimensions of 30, 40 and 50.

The experimental results of different inertia weight strategies on the given benchmark functions with various dimensions are presented in Tables 3, 4 and 5 which are with dimension  $D = 30, 40, 50$  respectively.

In Table 3,  $D = 30$ , the accuracies of the obtained re-

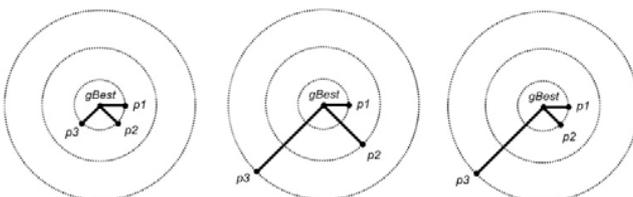


Fig. 1 States (compact, diverse and leap) of entropy-based gains.

**Table 2** Experimental settings of benchmark functions.

Test function (function id.)	Formula	Search space	Success threshold
Ackley ( $f_1$ )	$-20 \exp \left[ -\frac{1}{5} \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right] - \exp \left[ \frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right] + 20 + e$	$[-30, 30]^{30}$	5
Sphere ( $f_2$ )	$\sum_{i=1}^n x_i^2$	$[-100, 100]^{30}$	0.01
Rosenbrock ( $f_3$ )	$\sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	$[-30, 30]^{30}$	100
Rastrigin ( $f_4$ )	$\sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]_{30}$	100
Griewank ( $f_5$ )	$\frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^{30}$	1

**Table 3** Results of benchmark functions with D = 30.

		Precision (Deviation)			Time (Iterations)			Success (%)		
		LDIW	RIW	EGR-IW	LDIW	RIW	EGR-IW	LDIW	RIW	EGR-IW
$f_1$	Min	0.14241	0.050864	0.058659	798	376	90			
	Max	3.9474	18.4411	6.0805	967	>1000	>1000			
	Mean	2.2447	2.268	2.6273	886.36	599.267	140.213	100	90	94
	Std	0.86862	3.069	1.1365						
$f_2$	Min	0.042799	0.0019919	8.0404e-008	>1000	933	272			
	Max	5.1459	10.6588	76.2078	>1000	>1000	>1000			
	Mean	0.62946	0.71058	4.6399	>1000	953.667	315.90	0	6	42
	Std	0.9328	1.704	13.7899						
$f_3$	Min	119.4628	350.956	9.6023	>1000	>1000	283			
	Max	100485.898	4.8458e+06	2367.4692	>1000	>1000	>1000			
	Mean	7432.4778	198495.073	198.447	>1000	>1000	444.546	0	0	66
	Std	19173.554	814490.068	378.971						
$f_4$	Min	53.2161	34.9888	32.8404	817	277	91			
	Max	122.3318	301.1934	111.435	994	996	956			
	Mean	84.8738	91.6361	70.4781	928.333	735.263	242.478	84	76	92
	Std	17.3226	44.3716	18.0965						
$f_5$	Min	0.13319	0.010787	1.4588e-006	918	569	167			
	Max	1.0475	1.0959	1.6834	1000	995	249			
	Mean	0.52573	0.4271	0.25678	959.6735	798.435	196.326	98	92	86
	Std	0.27462	0.32966	0.43664						

sults with EGR-IW are better than those of LDIW and RIW on  $f_2$ ,  $f_3$ ,  $f_4$ , and  $f_5$ . In the end, EGR-IW also presented substantial advantages for the required iterations to reach the desired goals.

In 30-dimensional search space, the simulation results show that the EGR-IW performs better in comparison to the linearly decreasing inertia weight and random inertia weight widely used in many well-known test functions. Among all five test functions, EGR-IW obtained most of the advantages in terms of precision, success rates and convergent periods as compared to linearly decreasing and random inertia weight. Based on the analysis of experimental results, the EGR-IW has proved to be an appropriate adaptive mechanism providing effective control between exploration and exploitation in the process of particle swarm optimization.

In Table 4, D = 40, the accuracies of the obtained results with EGR-IW are better than those of LDIW and RIW

on all benchmark functions. The accuracies of EGR-IW are more close to the real optimum also with smaller deviations. EGR-IW also presented substantial advantages for the required iterations to reach the desired goals on all benchmark functions. It is remarkable that the EGR-IW outperforms overwhelmingly while the dimension of search space scale up from 30 to 40.

While the dimensions of search space are increased to 50, both the LDIW and RIW inertia strategies fail to reach the convergences on all benchmark functions. However, the EGR is still able to find the optimal solutions on Ackley and Griewank functions. The accuracies of EGR-IW are even better than that of LDIW and RIW with distinguish advantages. In addition to that, it is also observed that the accuracies on all benchmark functions are worse and worse on the increasing dimensions of search space.

**Table 4** Results of benchmark functions with D = 40.

		Precision (Deviation)			Time (Iterations)			Success (%)		
		LDIW	RIW	EGR-IW	LDIW	RIW	EGR-IW	LDIW	RIW	EGR-IW
<i>f<sub>1</sub></i>	Min	2.5828	3.8238	1.9366	918	874	168			
	Max	19.9573	20.6306	14.1024	>1000	>1000	>1000			
	Mean	7.8729	12.3153	4.3685	969.47	943	275.3889	30	8	72
	Std	4.5862	5.6656	2.3707						
<i>f<sub>2</sub></i>	Min	13.4687	7.7387	0.00131	>1000	>1000	168			
	Max	694.9262	104354.988	181.595	>1000	>1000	>1000			
	Mean	159.3519	6552.0795	21.7993	>1000	>1000	275.39	0	0	6
	Std	151.4572	17869.8815	44.5031						
<i>f<sub>3</sub></i>	Min	102411.0196	467100.271	24.0954	>1000	>1000	503			
	Max	161101518.2	430832622.76	2190.93	>1000	>1000	>1000			
	Mean	8521124.692	203864480.15	303.088	>1000	>1000	694.75	0	0	32
	Std	23496975.89	158815458.66	410.385						
<i>f<sub>4</sub></i>	Min	107.5702	124.7023	71.643	>1000	>1000	199			
	Max	321.6255	540.5723	180.090	>1000	>1000	>1000			
	Mean	181.9426	271.6627	119.183	>1000	>1000	409.375	0	0	16
	Std	44.2165	110.0536	23.4909						
<i>f<sub>5</sub></i>	Min	1.1212	1.0697	0.00427	>1000	>1000	244			
	Max	7.2543	940.1949	2.6451	>1000	>1000	>1000			
	Mean	2.4342	59.9687	0.859	>1000	>1000	310.8077	0	0	52
	Std	1.3631	160.8289	0.66077						

**Table 5** Results of benchmark functions with D = 50.

		Precision (Deviation)			Time (Iterations)			Success (%)		
		LDIW	RIW	EGR-IW	LDIW	RIW	EGR-IW	LDIW	RIW	EGR-IW
<i>f<sub>1</sub></i>	Min	6.81	7.94	2.54	>1000	>1000	239			
	Max	20.39	20.63	18.84	>1000	>1000	>1000			
	Mean	14.38	18.65	8.77	>1000	>1000	416	0	0	44
	Std	4.28	2.49	5.48						
<i>f<sub>2</sub></i>	Min	363.1144	485.73459	0.15947	>1000	>1000	>1000			
	Max	14418.5106	135406.2589	839.6896	>1000	>1000	>1000			
	Mean	3937.1459	53845.6497	123.7524	>1000	>1000	>1000	0	0	0
	Std	3771.6678	47422.4257	13.7889						
<i>f<sub>3</sub></i>	Min	4626285.72	129439777.7	107.7342	>1000	>1000	>1000			
	Max	575161621.2	589634445.9	526768942.6	>1000	>1000	>1000			
	Mean	217363306.1	417923955.0	23920528.0	>1000	>1000	>1000	0	0	0
	Std	180470223.1	112007972.7	98444663.4						
<i>f<sub>4</sub></i>	Min	178.8089	241.9318	116.2128	>1000	>1000	>1000			
	Max	813.3055	794.439	341.2708	>1000	>1000	>1000			
	Mean	375.7399	586.2425	187.3126	>1000	>1000	>1000	0	0	0
	Std	119.4924	166.2296	41.8542						
<i>f<sub>5</sub></i>	Min	4.268	5.3716	0.14929	>1000	>1000	373			
	Max	130.7666	1219.6563	8.2725	>1000	>1000	492			
	Mean	36.4343	485.6108	1.442	>1000	>1000	435.11	0	0	18
	Std	33.945	426.8018	1.1556						

4.2 Statistical Analysis

The differences between experimental results of EGR-IW and its corresponding opponents are tested by Wilcoxon sign rank test statistically to verify if there exist significant differences. Wilcoxon sign rank test is a nonparametric hypothesis test which has been applied to test solutions of global optimization [25], [26]. Wilcoxon sign rank test performs paired comparisons as an alternative to the paired

t-test without the assumption of normally distributed samples. Wilcoxon test rank the absolute value of differences of paired samples and then sum the signed ranks. Its results are summarized as R+, R-, which represent the sum of positive and negative ranks of samples in comparisons.

The results of Wilcoxon sign rank tests are computed by SPSS and listed in Tables 6, 7 and 8 with 30, 40 and 50 dimensions respectively whose acronyms, “LDIW-EGR” and “RIW-EGR”, represent the differences between the mean errors of traditional inertia weight strategies, Linear Decreases-

**Table 6** Wilcoxon signed rank test of mean error of accuracies regarding D = 30.

Function	Measures	LDIW-EGR	RIW-EGR
<i>f1</i>	R-	28	38
	R+	22*	12 <sup>#</sup>
	Z test	-1.588	-2.659
	p value	0.112	0.008
<i>f2</i>	R-	19*	18*
	R+	31	32
	Z test	-0.304	-0.024
	p value	0.761	0.981
<i>f3</i>	R-	4 <sup>#</sup>	0 <sup>#</sup>
	R+	46	50
	Z test	-5.884	-6.154
	p value	0.0004	0
<i>f4</i>	R-	15 <sup>#</sup>	15 <sup>#</sup>
	R+	35	35
	Z test	-3.557	-3.432
	p value	0.0004	0.001
<i>f5</i>	R-	9 <sup>#</sup>	13 <sup>#</sup>
	R+	41	37
	Z test	-3.470	-2.867
	p value	0.001	0.004

\* stands for accept, # stands for reject

**Table 8** Wilcoxon signed rank test of mean error of accuracies regarding D = 50.

Function	Measures	LDIW-EGR	RIW-EGR
<i>f1</i>	R-	8 <sup>#</sup>	2 <sup>#</sup>
	R+	42	48
	Z test	-5.034	-6.028
	p value	0.000	0.000
<i>f2</i>	R-	1 <sup>#</sup>	0 <sup>#</sup>
	R+	49	50
	Z test	-6.144	-6.154
	p value	0.000	0.000
<i>f3</i>	R-	1 <sup>#</sup>	0 <sup>#</sup>
	R+	49	50
	Z test	-5.816	-6.154
	p value	0.000	0.000
<i>f4</i>	R-	1 <sup>#</sup>	0 <sup>#</sup>
	R+	49	50
	Z test	-6.115	-6.154
	p value	0.000	0.000
<i>f5</i>	R-	1 <sup>#</sup>	0 <sup>#</sup>
	R+	49	50
	Z test	-6.144	-6.154
	p value	0.000	0.000

\* stands for accept, # stands for reject

**Table 7** Wilcoxon signed rank test of mean error of accuracies regarding D = 40.

Function	Measures	LDIW-EGR	RIW-EGR
<i>f1</i>	R-	10 <sup>#</sup>	7 <sup>#</sup>
	R+	40	43
	Z test	-4.494	-5.584
	p value	0.000	0.000
<i>f2</i>	R-	4 <sup>#</sup>	3 <sup>#</sup>
	R+	46	47
	Z test	-5.276	-5.691
	p value	0.000	0.000
<i>f3</i>	R-	0 <sup>#</sup>	0 <sup>#</sup>
	R+	50	50
	Z test	-6.154	-6.154
	p value	0.000	0.000
<i>f4</i>	R-	4 <sup>#</sup>	1 <sup>#</sup>
	R+	46	49
	Z test	-5.913	-6.115
	p value	0.000	0.000
<i>f5</i>	R-	4	3
	R+	46	47
	Z test	-5.671.	-5.874
	p value	0.000	0.000

\* stands for accept, # stands for reject

ing (LDIW) and Random (RIW), in comparisons to EGR-IW. It is important to remind that the *p* value is computed based on the less one of numbers of positive and negative ranks. Therefore, the decisions of Wilcoxon signed rank test are made according to the *p* value from the less one of R+ and R-. Besides, the conclusions of significant small are desired since that the given benchmark functions are designed

to locate the minimum value.

Table 6 presents results of Wilcoxon signed rank test of EGR-IW in comparisons to the others regarding mean error of accuracies as the performance criteria on given benchmark functions in 30-dimensional search space. Table 6 shows that EGR-IW outperforms LDIW and RIW inertia strategy in most of the given benchmark functions except the *f1*\_RIW, *f2*\_LDIW and *f2*\_RIW.

The results for 40-dimensional and 50-dimensional search space are shown in Tables 7 and 8 respectively. The analysis show that EGR inertia strategies take overwhelming advantages and perform significantly better than LDIW and RIW on all the benchmark functions. EGR-IW is therefore regarded significantly better than LDIW and RIW.

### 4.3 Analysis on Adaptability

The changes of entropy-based gain in the evolutions of PSO are plotted and analyzed on adaptability. The figures of EG on iterations show the various types of turbulence corresponding to the distinct test functions. In Figs. 2 and 3, both Ackley and Sphere are unimodal functions that contain no local optima, the global optimum is reached before the half of iterations. The remaining high EG value implies most particles have come together and become equidistant around the global best location.

In Fig. 4, the Rosenbrock function shows that the exploration and exploitation movements appearing staggered in the middle period (400–600 of 1000 iterations) and the ending period (850–1000 iterations) of the evolution. The Rosenbrock function is also a unimodal function whose gradient is relatively small on its wide flat valley. Therefore, a

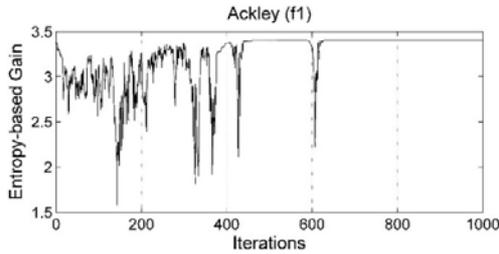


Fig. 2 Entropy-based gain on Ackley function.

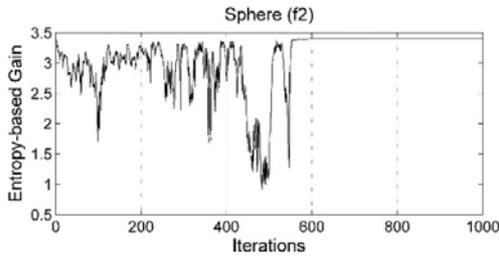


Fig. 3 Entropy-based gain on Sphere function.

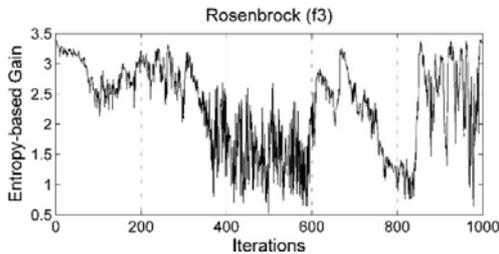


Fig. 4 Entropy-based gain on Rosenbrock function.

little improvement of global best is likely to produce a significant change in location which might cause huge changes in EG value. The far-moved tiny-improved global best generate frequent shifts between explorations and exploitations.

The benchmark functions in Figs. 5 and 6 are multimodal. The turbulences between exploration and exploitation moves are clearly observed. The changing between explorations and exploitations shown in the Rastrigin test produce the perturbed effects on inertia weight is considered to be a valuable feature of the EGR that can attain high quality results even under a fast convergence speed. From the frequently changing EG, the perturbed EGR adjusted inertia weight is considered has the capacity to escape from local optima for a good quality.

In general, the blending of exploration and exploitation movements result in adaptive tunings corresponding to the EG value that empirically enhance the adaptability of inertia weight strategies against the varying states of evolution. According to the demonstrations from the figures of EG, the proposed EGR inertia weight strategy provides an appropriate adaptive mechanism reflecting the evolutionary state of a given particle swarm. Consequently, EG is regarded as an appropriate measure for an adaptive control that tuning balances between explorations and exploitations for a particle swarm optimizer.

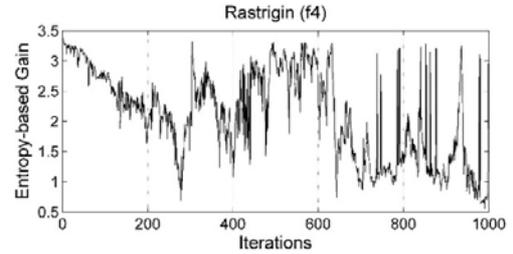


Fig. 5 Entropy-based gain on Rastrigin function.

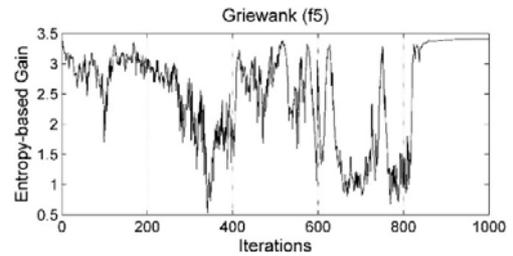


Fig. 6 Entropy-based gain on Griewank function.

## 5. Conclusions

In this work, we proposed the entropy-based gain (EG) or entropy-based gain regulator (EGR) to improve performance of the original particle swarm optimization. The mathematical term EGR is derived from the idea of entropy based on the concept of the measure of disorder of a system, originally in the thermodynamic context. We further investigated the performance of EGR applied empirically to linear decreasing inertia weight and analyzed observed data to illuminate the effect of the EGR and its relations to inertia weight strategies. The performance results from introducing the EGR to the linear decreasing inertia weight is quite good. Entropy-based gain is confirmed to be an effective parameter for detecting and providing feedback for adjusting the inertia weight adaptively in PSO algorithms for controlling the balancing of the exploration and exploitation of searches.

In this study, the EGR provides effective tuning of inertia weight using the distances between particles and the global best. Though the EGR has proved to be able to produce significant improvements of PSO on the selected five test functions, a further analysis would be valuable to clarify the correlations of location distances between particles from the global best on the search space and the differences of fitness value on the solution space. Investigating strategies other than linear decreasing inertia weight is also encouraged.

In addition, we expect further investigations to discover correlations between the EGR and such features as swarm size, number of dimensions, different limits on generations, and diverse kinds of problems. To clarify the subtle and sophisticated effects of the EGR, the tests on the rotated and shifted functions are worth delicate experiments to pro-

vide further insights in the future. We expect that studies on the relations between particle trajectories and entropy-based gain will be especially interesting and valuable.

## Acknowledgments

The research is based on work supported by Taiwan Ministry of Science and Technology under Grant No. MOST 103-2410-H-006-055-MY3.

## References

- [1] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proc. IEEE International Conference on Neural Networks, ICNN '95, vol.4, pp.1942–1948, 1995.
- [2] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," Proc. IEEE International Conference on Evolutionary Computation (IEEE World Congress on Computational Intelligence), pp.69–73, 1998.
- [3] Y. Shi and R.C. Eberhart, "Parameter selection in particle swarm optimization," Evolutionary Programming VII, Lecture Notes in Computer Science, vol.1447, pp.591–600, Springer, Berlin, Heidelberg, 1998. .
- [4] Y. Shi and R.C. Eberhart, "Empirical study of particle swarm optimization," Proc. 1999 Congress on Evolutionary Computation, CEC 99, 1999
- [5] I.C. Trelea, "The particle swarm optimization algorithm: Convergence analysis and parameter selection," Inform. Process. Lett., vol.85, no.6, pp.317–325, 2003.
- [6] X.-S. Yang, "Metaheuristic optimization: Algorithm analysis and open problems," Experimental Algorithms, Lecture Notes in Computer Science, vol.6630, pp.21–32, Springer, 2011.
- [7] O. Olorunda and A.P. Engelbrecht, "Measuring exploration/exploitation in particle swarms using swarm diversity," Proc. IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), CEC 2008, pp.1128–1134, 2008.
- [8] J.C. Bansal, P.K. Singh, M. Saraswat, A. Verma, S.S. Jadon, and A. Abraham, "Inertia weight strategies in particle swarm optimization," 2011 Third World Congress on Nature and Biologically Inspired Computing (NaBIC), pp.633–640, 2011.
- [9] J.F. Schutte and A.A. Groenwold, "A study of global optimization using particle swarms," Journal of Global Optimization, vol.31, no.1, pp.93–108, 2005.
- [10] A. Nickabadi, M.M. Ebadzadeh, and R. Safabakhsh, "A novel particle swarm optimization algorithm with adaptive inertia weight," Applied Soft Computing, vol.11, no.4, pp.3658–3670, 2011.
- [11] G. Chen, X. Huang, J. Jia, and Z. Min, "Natural exponential inertia weight strategy in particle swarm optimization," The Sixth World Congress on Intelligent Control and Automation, WCICA 2006, pp.3672–3675, 2006.
- [12] W. KaiJie and W. JianXin, "Dynamic analysis and optimization of parallel manipulator," International Journal of Computer Science and Application, vol.3, no.1, pp.42–45, 2014.
- [13] T.O. Ting, Y. Shi, S. Cheng, and S. Lee, "Exponential inertia weight for particle swarm optimization," Advances in Swarm Intelligence, Lecture Notes in Computer Science, vol.7331, pp.83–90, Springer, Berlin, Heidelberg, 2012.
- [14] A. Banks, J. Vincent, and C. Anyakoha, "A review of particle swarm optimization Part I: Background and development," Natural Computing, vol.6, no.4, pp.467–484, 2007.
- [15] R.C. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms," Proc. 2001 Congress on Evolutionary Computation, pp.94–100, 2001.
- [16] J.-B. Park, Y.-W. Jeong, J.-R. Shin, and K.Y. Lee, "An improved particle swarm optimization for nonconvex economic dispatch problems," IEEE Trans. Power Syst., vol.25, no.1, pp.156–166, 2010.
- [17] Y. Feng, G.-F. Teng, A.-X. Wang, and Y.-M. Yao, "Chaotic inertia weight in particle swarm optimization," Second International Conference on Innovative Computing, Informatio and Control (ICICIC 2007), p.475, 2007.
- [18] C. Dong, G. Wang, Z. Chen, and Z. Yu, "A method of selfadaptive inertia weight for PSO," 2008 International Conference on Computer Science and Software Engineering, pp.1195–1198, 2008.
- [19] M.S. Arumugam, M.V.C. Rao, and A. Chandramohan, "A new and improved version of particle swarm optimization algorithm with global-local best parameters," Knowledge and Information Systems, vol.16, no.3, pp.331–357, 2008.
- [20] B.K. Panigrahi, V.R. Pandi, and S. Das, "Adaptive particle swarm optimization approach for static and dynamic economic load dispatch," Energy Conversion and Management, vol.49, no.6, pp.1407–1415, 2008.
- [21] B. Liu, L. Wang, Y.-H. Jin, F. Tang, and D.-X. Huang, "Improved particle swarm optimization combined with chaos," Chaos, Solitons & Fractals, vol.25, no.5, pp.1261–1271, 2005.
- [22] C.E. Shannon and W. Weaver, A Mathematical Theory of Communication, American Telephone and Telegraph, 1948.
- [23] X. Yang, J. Yuan, J. Yuan, and H. Mao, "A modified particle swarm optimizer with dynamic adaptation," Applied Mathematics and Computation, vol.189, no.2, pp.1205–1213, 2007.
- [24] J.M. Dieterich and B. Hartke, "Empirical review of standard benchmark functions using evolutionary global optimization," arXiv preprint arXiv:1207.4318, 2012.
- [25] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," Journal of Machine Learning Research, vol.7, pp.1–30, 2006.
- [26] 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.



**Hei-Chia Wang** is a professor in the Institute of Information Management at National Cheng Kung University, Taiwan. His research focuses on knowledge discovery, text mining, e-learning and bioinformatics. Wang obtained both M.Sc. in information system engineering and Ph.D. in informatics from the University of Manchester (UMIST), UK.



**Che-Tsung Yang** is currently a doctoral student in the Institute of Information Management at National Cheng Kung University in Taiwan. His research interests include machine learning, and heuristic optimization.