

# An Investigation of Logarithm Decreasing Inertia Weight Particle Swarm Optimization in Global Optimization Problem

Murinto  
Department of Informatic  
Engineering  
Universitas Ahmad Dahlan  
Yogyakarta, Indonesia

Adhi Prahara  
Department of Informatic  
Engineering  
Universitas Ahmad Dahlan  
Yogyakarta, Indonesia

Erik Iman Heri Ujjianto  
Department of Informatic  
Engineering  
Universitas Teknologi Yogyakarta,  
Indonesia

## ABSTRACT

This research investigates Logarithm Decreasing Inertia Weight (LogDIW) to improve the performance of Particle Swarm Optimization (PSO). The general problem of PSO algorithm is premature convergence when solving complex optimization problem. Some researchers try to solve the problem by modifying the PSO or proposing another PSO variants. Some PSO variants proved to have a better performance than the original PSO. The purpose of this research is to obtain some experimental facts to prove the efficiency of LogDIWPSO if the parameters are tuned correctly and to show that the LogDIWPSO performs better compared to the other PSO variants. In the early step of the experiment, a percentage value of search space boundary is obtained. This step is important to compute the velocity threshold of LogDIW based on the optimization problem. The next experiment is done to measure the performance of LogDIWPSO using six benchmark functions in optimization problems and to prove the superiority of LogDIWPSO compared to the other PSO variants. The experiment result shows that LogDIW achieves better performance than the other PSO variants.

## General Terms

Algorithms, Optimization Problems

## Keywords

Inertia weight, particle swarm optimization, logarithm decreasing inertia weight

## 1. INTRODUCTION

Since introduced in 1995, Particle Swarm Optimization (PSO) had been improved and widely used in many applications. A modified PSO mainly focus to improve the convergence in order not to be trapped in the local optima. Some modifications also done in the parameters such as in the inertia weight, velocity clamping, the change in cognitive and social aspect value, and in the determination of personal best position (Pbest) and the global best position (Gbest).

Inertia weight plays important role in the trade-off process between the diversification and intensification ability of PSO algorithm. When inertia weight strategy implemented in PSO algorithm, the particles will move around and adjust their velocity and position according to the original PSO equation in the search space. Inertia weight also has a role to balance the exploration and exploitation. Inertia weight will determine the contribution of the previous particle's velocity for the current velocity. Original PSO that introduced by Kennedy and Eberhart in 1995 did not have inertia weight [1]. The concept of inertia

weight was introduced in 1998 by Shi and Eberhart [2] [3]. In their research, a constant inertia weight was included in the algorithm. A large value of inertia weight facilitates the global search while a small value of inertia weight facilitates the local search. Therefore, the best result of PSO algorithm can be achieved by choosing the correct inertia weight. Several studies using inertia weight in PSO were carried out by [4][5][6].

Dynamic adjustment of inertia weight to improve the PSO performance have been proposed by many researchers. One of them is Random Inertia Weight (RIW) [7]. The experiment proved that this strategy will improve the convergence at the earlier iteration of the algorithm. Linearly Decreasing Inertia Weight (LDIW) strategy [8] also proposed to improve the efficiency and performance of PSO. The experiment shows that the inertia weight from 0.9 up to 0.4 produces the best result.

Fang et al. (2008) proposed Chaotic Inertia Weight that took the advantage of chaos optimization. The research compares Chaotic Random Inertia Weight (CRIW)-PSO and Random Inertia Weight (RIW)-PSO and resulted in the better performance of CRIW-PSO [9]. The algorithm performs rough search step and fine search step alternately on its all evolutionary process. Malik et al. (2007) presented a Sigmoid Increasing Inertia Weight (SIIW) [10]. They find that sigmoid function contributes in obtaining temporary minimum fitness function while Linearly Increasing Inertia Weight (LDIW) contributes to a faster convergence. Therefore, they combine the sigmoid function and Linearly Increasing Inertia Weight and also introduce SIIW that produces a significant improvement in the aspect of fast convergence and narrowing the aggressive movement to the solution region. Oscillating Inertia Weight [11] proposed a periodic change between the global search and local search and proved that the strategy is competitive in some cases especially in the aspect of convergence speed.

Gao et al. in 2008 proposed a Logarithmic Decreasing Inertia Weight (LogDIW) with chaos mutation operator. Chaos mutation operator able to increase the ability to jump out from the premature convergence and increase the convergence speed and accuracy [12]. To handle the stagnation and premature convergence from PSO algorithm, Gao et al. in 2009 proposed Exponent Decreasing Inertia Weight (e-DIW) with stochastic mutation (SM) [13]. Stochastic mutation (SM) is used to increase the diversity of swarm while the e-DIW is used to increase the personal convergence speed. The larger inertia weight facilitates at the inclusive phase of the search space and then decrease linearly into smaller inertia weight. In [14] an improved particle swarm optimization (EWPSO) have proposed a novel strategy for inertia weight. In this algorithm, a nonlinear

inertia weight used. An exponential function of maximal and minimal fitness used this new inertia in each iteration. The result was compared with the standard PSO with linear decreasing inertia weight and RNW-PSO. Form the simulation showed that EWPSO more effective and efficient for the benchmark test.

Based on the idea from [9], this research will investigate the used of Logarithm Decreasing Inertia Weight in PSO. The study will cover the comparison of the best fitness value and the speed of convergence between this algorithm and the other PSO variants. This paper is organized into six sections. Section 1 introduces the problems and the reason why logarithm decreasing inertia weight is proposed to handle some problems in PSO. Section 2 explains the strategy of particle swarm optimization and the review of seven inertia weight strategies of particle swarm optimization will be explained in Section 3. Section 4 explains the experiment scenario and discusses the result in detail. The conclusion of this work is presented in Section 5.

## 2. PARTICLE SWARM OPTIMIZATION

PSO was introduced by Eberhart and Kennedy in 1995 then modified by Shi and Eberhart in 1998 by adding a constant inertia weight in the algorithm. Large inertia weight facilitates global search while small inertia weight facilitates local search. The original PSO algorithm performs velocity updated and position updated (Everhart and Kennedy, 1995) as shown in equation (1) and equation (2) respectively.  $v_{id}^{t+1} = w \cdot v_{id}^t + c_1 \cdot r_1 \cdot (P_{id}^t - x_{id}^t) + c_2 \cdot r_2 \cdot (P_{gd} - x_{id}^t)$  (1)  
 $x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1}$  (2)

Where  $c_1$  and  $c_2$  are positive constants called acceleration coefficients,  $r_1$  and  $r_2$  are random numbers in the range of [0, 1], and  $w$  is the inertia weight. Large inertia weight will facilitate global exploration while small inertia weight will facilitate local exploitation. The  $i$ -th particle is presented as  $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$ . The best previous position of the  $i$ -th particle is stored and presented as  $P_i = (p_{i1}, p_{i2}, \dots, p_{id})$ . The position gives the best fitness value. Index of the best particle in the population is presented with  $g$  symbol. The velocity of the  $i$ -th particle presented as  $V_i = (v_{i1}, v_{i2}, \dots, v_{id})$ . During the update, the particle's velocity in each dimension is bounded to  $v_{max}$ .  $D$  is a dimension of the search space.

## 3. A REVIEW OF INERTIA WEIGHT

This study covers some PSO variants namely: (1) Constant Inertia Weight (CIW), (2) Random Inertia Weight (RIW), (3) Linearly Decreasing Inertia Weight (LDIW), (4) Chaotic Decreasing Inertia Weight (CDIW), (5) Chaotic Random Inertia Weight (CRIW), (6) Logarithm Decreasing Inertia Weight (LogDIW).

### 3.1. Constant Inertia Weight

A modified PSO with the concept of inertia weight  $w$  was introduced for the first time by Shi and Eberhart [2]. The research used a constant inertia weight (CIW) where a large constant is suitable for exploration while a small constant is suitable for exploitation. CIW can be computed using equation (5).

$$w^t = constant \quad (3)$$

### 3.2. Random Inertia Weight

Randomize in PSO was introduced by Eberhart and Shi [3] through the inertia weight strategy. A new method to compute the inertia weight is by using particle swarm to search and

optimize the dynamic system. The random value of inertia weight is computed using equation (4).

$$\omega_t = 0.5 + \frac{rand(.)}{2} \quad (4)$$

Where  $rand(.)$  is the random value in the range of 0 and 1. Equations

### 3.3. Linearly Decreasing Inertia Weight

LDIW-PSO is a PSO variant that implements linear descending (decreasing) inertia weight [4]. The strategy improves the algorithm significantly. In LDIW, inertia weight starts from some initial values and linearly decrease to smaller final values. Large inertia weight facilitates global search while small inertia weight facilitates local search.

The initial and final values that usually used are 0.9 or 0.4. High initial value makes the particle moves in low viscosity medium that facilitates exploration while small inertia value makes particles moves in high viscosity medium that facilitates exploitation. However, by using linearly decreasing inertia weight lead PSO to premature convergence which is a fast convergence to the early optimum position. Equation (3) shows how to compute the LDIW.

$$\omega_t = (\omega_{start} - \omega_{end}) \left( \frac{T_{max} - t}{T_{max}} \right) + \omega_{end} \quad (3)$$

Where  $\omega_{start}$  and  $\omega_{end}$  are the initial value and final value of the inertia weight,  $t$  is the iteration at run time,  $T_{max}$  is the number of maximum iteration, and  $\omega_t \in [0, 1]$  is the value of inertia weight at  $t$ -th iteration.

### 3.4. Chaotic Descending Inertia Weight

The idea of Chaotic Inertia Weight (CIW) uses chaotic mapping to find the inertia weight coefficients. Feng et al. [6] proposed to use logistic mapping. Chaotic inertia weight strategy consists of Chaotic Descending Inertia Weight (CDIW) and Chaotic Random Inertia Weight (CRIW). CDIW strategy uses the advantages of chaos optimization. Chaos is a non-linear dynamic system that sensitive to the initial value. The system has ergodicity and stochastic property characteristic. The final goal is to handle the premature convergence that occurred in LDIWPSO. Equation (6) presents the chaotic descending inertia weight.

$$z_{k+1} = \mu * z_k * (1 - z_k) \quad (5)$$

Where  $\mu = 4$  and  $z_k$  is the  $k$ -th chaotic number. The generator map has value between 0 and 1, initial value of  $z_0 \in (0, 1)$  and  $z_0 \notin (0, 0.25, 0.5, 0.75, 1.0)$ .

$$\omega_t = (\omega_{start} - \omega_{end}) \left( \frac{T_{max} - T}{T_{max}} \right) + \omega_{end} * z_{k+1} \quad (6)$$

Where  $\omega_{start}$  and  $\omega_{end}$  is the initial and final inertia weight. CDIWPSO shows a better convergence precision, faster convergence speed and better global search performance.

### 3.5. Chaotic Random Inertia Weight

Chaotic Random Inertia Weight (CRIW) was introduced by Feng et al [6] and the formula is shown in equation (7). The purpose of this strategy is to improve the random inertia weight in equation (4) using logistic map in equation (5) and to avoid the local optima in the search process using the advantages of chaotic optimization.

$$\omega_t = \frac{rand(.)}{2} + 0.5 * z_{k+1} \quad (7)$$

Where  $rand(.)$  is uniform random number in the range of [0, 1].

### 3.6. Logarithm Decreasing Inertia Weight

The empirical study shows that PSO with large value of inertia weight ( $w$ ) has better global search capability compared to the smaller  $w$  with faster convergence. Gao et al. [9] introduce Logarithm Decreasing Inertia Weight (LogDIW) as shown in equation (8).

$$w = w_{max} + (w_{min} - w_{max}) \times \log_{10}(a + 10t / T_{max}) \quad (8)$$

Where  $a$  is constant to adjust the evolutionary speed,  $a = 1$ .

### 4. EXPERIMENT RESULT

The claim of this research is validated using various experiments. Experiment is done by comparing the Logarithm Decreasing Inertia Weight Particle Swarm Optimization (LogDIWPSO) with the other five variants of PSO namely Chaotic Descending Inertia Weight Particle Swarm Optimization (CDIWPSO): Constant Inertia Weight Particle Swarm Optimization (CIWPSO), Random Inertia Weight Particle Swarm Optimization (RIWPSO), Linear Decreasing Inertia Weight Particle Swarm Optimization (LDIWPSO), and Chaotic Random Inertia Weight Particle Swarm Optimization (CRIWPSO).

Six benchmark functions from the literature are used to test the performance of CDIW-PSO, CIW-PSO, RIW-PSO, LDIW-PSO, ChiW-PSO, CRIW-PSO, LogDIW-PSO, and e-DIW-PSO. Those benchmarks are chosen because its combination can be used to validated various kinds of PSO. The scenario of experiment involves different setting of parameters. Some parameter values in this research  $c_1 = c_2 = 2.0$ ,  $w_{max} = 0.9$ ,  $w_{min} = 0.4$ ,  $V_{min} = \alpha X_{min}$  and  $V_{max} = \alpha X_{max}$ , where  $\alpha = 0.1$ .

The benchmark functions can be classified into two groups namely unimodal and multimodal functions [15]. The benchmark functions that used in this research are Sphere Function (F1), Rastrigin Function (F2), Griewank Function (F3), Schaffer Function (F5), Ackley Function (F5), and Rosenbrock Function (F6) as shown in Table 1. The purpose of this experiment is to compare LogDIWPSO with the other five PSO variants. The six benchmark functions use dimension of 30 except Chaffer function which use dimension of 2. The number of maximum iteration is 1500 with swarm size is 20. The experiment is done 30 times. The goal of this experiment is to prove that LogDIWPSO is more efficient compared with the other PSO variants.

#### Sphere Function

Figure 1 shows the comparison of mean value of best fitness against the number of iteration. The mean value of Best Fitness, Worst Fitness, Mean and Deviation Standard of Sphere function for seven inertia weight based PSO variants is shown in Table 1. Although the CRIWPSO and LogDIWPSO show a fast convergence in the early search stage, they are a little behind the other five inertia weights after approximately 200 iterations. The final solution of LogDIWPSO and the other six inertia weights after approximately 800 iterations is similar.

Figure 3 shows the comparison of mean value of best fitness against the number of iteration. The mean value of Best Fitness, Worst Fitness, Mean and Deviation Standard of Griewank

function of seven inertia weight based PSO variants is shown in Table 1. Although the CRIW-PSO and LogDIW-PSO show a fast convergence in the early search stage, they are a little behind the other five inertia weights after approximately 150 iterations. The final solution of LogDIW-PSO, CRIW-PSO and CIW-PSO after approximately 800 iterations is similar.

Table 1. The Six Benchmark functions

Name	Expression and Condition
Sphere function	$f_1(x) = \sum_{i=1}^n x_i^2$ , $n=30$ $x_i \in [-100,100]$ , $v_{max} = 100$ , $\min(f_1) = f_1(0, \dots, 0) = 0$
Rastrigin function	$f_2(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$ , $n=30$ $x_i \in [-600,600]$ , $v_{max} = 600$ , $\min(f_2) = f_2(0, \dots, 0) = 0$
Griewank function	$f_3(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ , $n=30$ $x_i \in [-600,600]$ , $v_{max} = 600$ , $\min(f_3) = f_3(0, \dots, 0) = 0$
Schaffer's function	$f_4(x) = \sum_{i=1}^{d-1} \left( 0.5 + \frac{\sin^2\left(\sqrt{x_{i+1}^2 + x_i^2}\right)}{(0.001 + (x_{i+1}^2 + x_i^2))^2} \right)$ , $n=30$ , $x_i \in [-100,100]$ , $v_{max} = 100$ , $\min(f_4) = f_4(0, \dots, 0) = 0$
Ackley function	$f_5(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) + 20 + e$ , $n=30$ $x_i \in [-32,32]$ , $v_{max} = 32$ , $\min(f_5) = f_5(0, \dots, 0) = 0$
Rosenbrock function	$f_6(x) = \sum_{i=1}^{d-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$ , $n=30$ $x_i \in [-100,100]$ , $v_{max} = 100$ , $\min(f_6) = f_6(0, \dots, 0) = 0$

#### Griewank Function

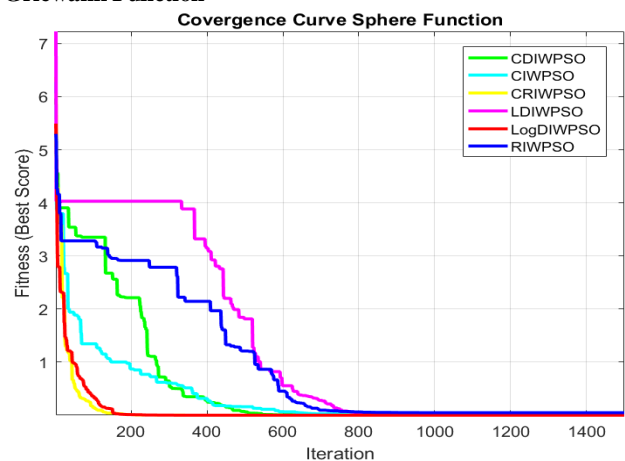


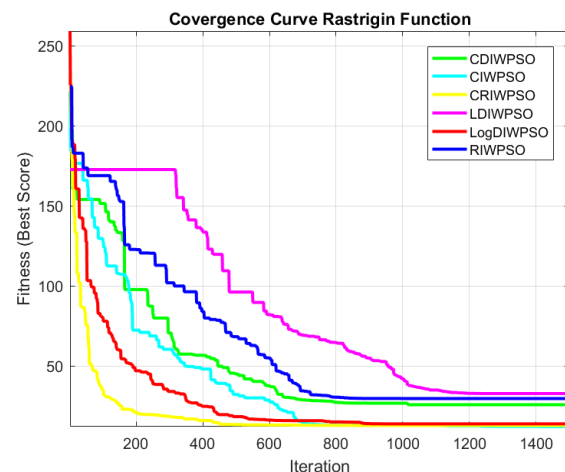
Fig. 1: Comparison between the best fitness values of Sphere function to the number of iteration

**Table 2. Comparison of best fitness, worst fitness, and mean and deviation standard value of seven PSO variants based on inertia weight on six benchmark functions**

Inertia Weight PSO	Performance Index	Inertia Weight PSO					
		Sphere	Rastrigin	Griewank	Schaffer's	Ackley	Rosenbrock
CDIWPSO	Best Fitness	1,0370E-23	2,7859E+01	3,3307E-16	0,0000E+00	1,6462E+00	7,7156E+01
	Mean Fitness	6,0860E-25	1,8274E+01	3,3307E-17	0,0000E+00	5,4874E-02	2,5185E+01
	Deviation Standard	1,9291E-24	4,0266E+00	8,8194E-17	0,0000E+00	3,0056E-01	1,7703E+01
	Worst Fitness	1,8547E-29	1,2934E+01	0,0000E+00	0,0000E+00	4,4409E-15	2,1481E+00
RIWPSO	Best Fitness	2,4387E-01	1,0039E+02	1,2511E-02	9,2174E-02	5,5774E-01	1,3520E+02
	Mean Fitness	1,3807E-01	6,5914E+01	5,2350E-03	1,6327E-02	3,7202E-01	6,2797E+01
	Deviation Standard	5,3489E-02	1,4330E+01	2,6764E-03	1,5767E-02	1,1259E-01	2,8596E+01
	Worst Fitness	3,3440E-02	3,4124E+01	2,2706E-03	7,6228E-03	1,7466E-01	3,5634E+01
LDIWPSO	Best Fitness	7,2414E-09	8,5566E+01	2,1474E-02	3,8952E-02	4,2572E-05	8,0315E+01
	Mean Fitness	4,6305E-10	5,0444E+01	9,6232E-04	1,5866E-02	5,9092E-06	2,6547E+01
	Deviation Standard	1,4926E-09	1,2455E+01	4,1023E-03	8,7258E-03	9,3318E-06	1,0402E+01
	Worst Fitness	1,2239E-12	3,3829E+01	1,4433E-14	5,4916E-03	4,2102E-07	1,7396E+01
CIWPSO	Best Fitness	9,7898E-02	7,6712E+01	7,4310E-03	2,5962E-02	6,5311E-02	8,5450E+01
	Mean Fitness	3,4953E-02	3,9595E+01	5,7337E-04	1,2784E-02	3,2867E-02	3,2734E+01
	Deviation Standard	2,1221E-02	1,1424E+01	1,8626E-03	6,3775E-03	1,1635E-02	1,5812E+01
	Worst Fitness	5,0652E-03	2,4192E+01	1,8920E-05	2,2256E-03	1,4385E-02	2,3945E+01
CRIWPSO	Best Fitness	6,0332E-14	7,9596E+01	1,1467E-01	3,5814E-02	2,2210E+00	8,1771E+01
	Mean Fitness	3,7870E-15	4,9449E+01	3,8222E-03	1,6363E-02	6,8442E-01	2,9496E+01
	Deviation Standard	1,1332E-14	1,5403E+01	2,0935E-02	7,7067E-03	7,6111E-01	1,6906E+01
	Worst Fitness	9,5795E-19	2,0894E+01	0,0000E+00	6,5614E-03	7,7251E-10	1,7603E+01
LogDIWPSO	Best Fitness	6,4000E-14	3,0844E+01	2,0317E-14	2,2256E-03	2,0133E+00	7,9851E+01
	Mean Fitness	4,3969E-15	2,2950E+01	1,9355E-15	1,1145E-04	3,9915E-01	1,1533E+01
	Deviation Standard	1,4829E-14	3,2732E+00	3,7561E-15	4,4838E-04	7,4211E-01	2,5302E+01
	Worst Fitness	4,1195E-19	1,6914E+01	0,0000E+00	0,0000E+00	2,5788E-06	6,6449E-03

### Rastigrin Function

Figure 2 shows the comparison of mean value of best fitness against the number of iteration. The mean value of Best Fitness, Worst Fitness, Mean and Deviation Standard of Rastrigin function of seven inertia weight based PSO variants is shown in Table 1. Although the CRIW-PSO and LogDIW-PSO show a fast convergence in the early search stage, they are a little behind the other five inertia weights after approximately 500 iterations. The final solution of LogDIW-PSO, CRIW-PSO and CIW-PSO after approximately 700 iterations is similar.



**Fig. 2: Comparison between the best fitness values of Rastrigin function to the number of iteration**

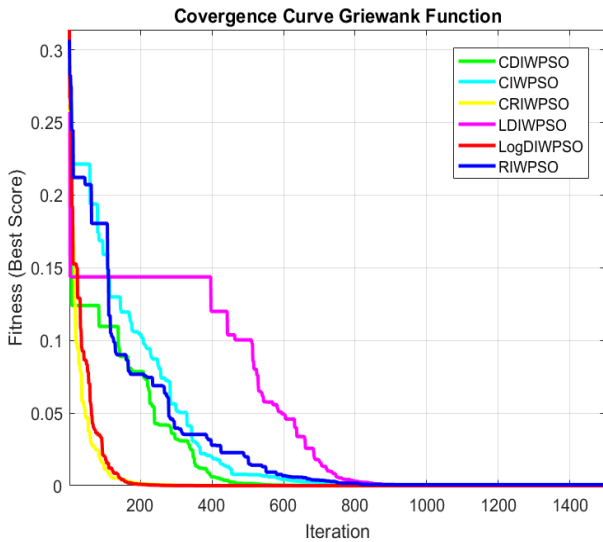


Fig. 3: Comparison between the best fitness values of Griewank to against the number of iteration.

### Schaffer Function

Figure 4 shows the comparison of mean value of best fitness against the number of iteration. The mean value of Best Fitness, Worst Fitness, Mean and Deviation Standard of Schaffer function of seven inertia weight based PSO variants is shown in Table 1. Although the CRIW-PSO and LogDIW-PSO show a fast convergence in the early search stage, they are a little behind the other five inertia weight after approximately 150 iterations. The final solution of LogDIW-PSO and CRIW-PSO after approximately 400 iterations is similar.

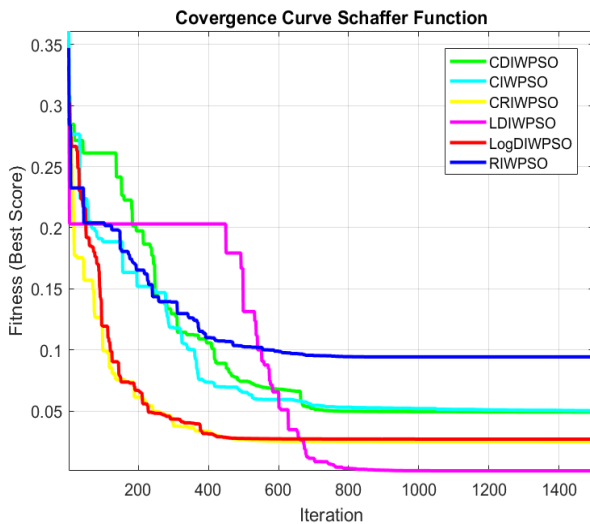


Fig. 4: Comparison between the best fitness values of Schaffer function to the number of iteration.

### Ackley Function

Figure 5 shows the comparison of mean value of best fitness against the number of iteration. The mean value of Best Fitness, Worst Fitness, Mean and Deviation Standard of Ackley function of seven inertia weight based PSO variants is shown in Table 1. Although the CRIWPSO and LogDIWPSO show a fast convergence in the early search stage, they are a little behind the other five inertia weights after approximately 180 iterations. The final solution of LogDIWPSO, CDIW and CRIWPSO for approximately 1000 iterations is similar.

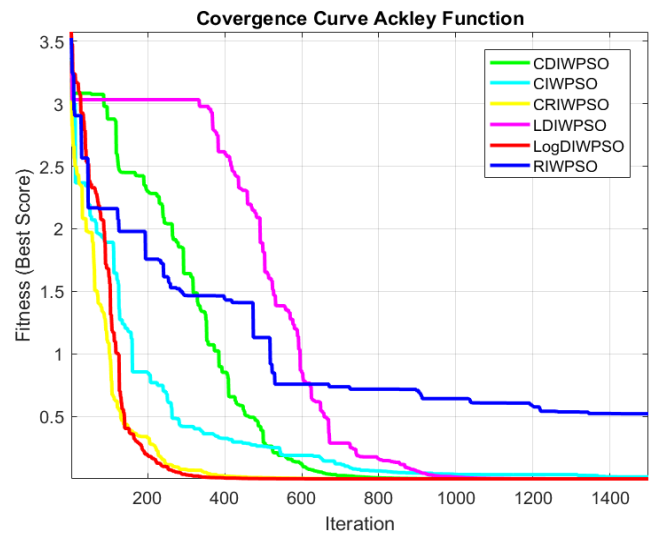


Fig. 5: Comparison between the best fitness value of Ackley function to the number of iteration.

### Rosenbrock Function

Figure 6 shows the comparison of mean value of best fitness against the number of iteration. The mean value of Best Fitness, Worst Fitness, Mean and Deviation Standard of Rosenbrock function of seven inertia weight based PSO variants is shown in Table 1. Although the CRIW-PSO and LogDIW-PSO show a fast convergence in the early search stage, they are a little behind the other five inertia weights after more than 300 iterations. The final solution of LogDIW-PSO, LDIW and CRIW-PSO after more than 900 iterations is similar.

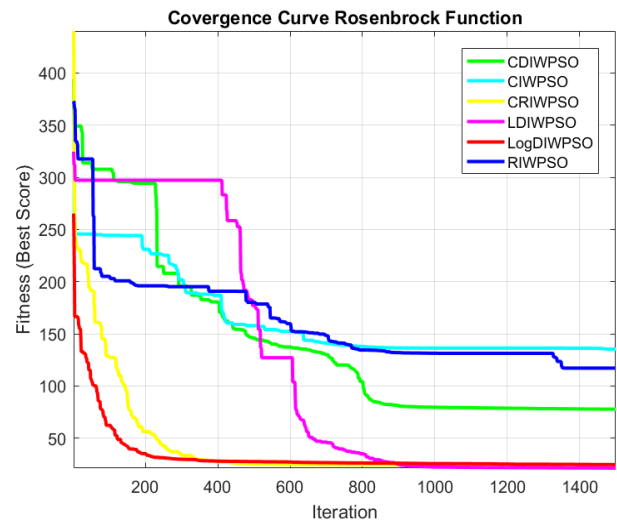


Fig. 6: Comparison between the best fitness value of Rosenbrock function to the number of iteration.

From the result of experiments that have been carried out using the benchmark function, it can be seen that the convergence of LogDIWPSO algorithm is better than other PSO variants. This can be seen in Fig.2 – Fig.6.

## 5. ACKNOWLEDGMENTS

Thank you to the Kemenristek Dikti Indonesia who has provided funds to conduct this research in 2021.

## 6. CONCLUSION

In this study, a strategy of inertia weight is proposed using a

logarithmic decreasing inertia weight (LogDIW). In this paper the use of a decreasing inertia weight strategy in several variants of PSO also describe. The experimental results show that the strategy using LogDIW on PSO convergence is better than other linear strategy during early stages of the search process. In various optimization problems tested on commonly used benchmark functions, this strategy has better performance than other linear inertia weight strategies.

The proposed algorithm LogDIW PSO can be applied to image processing such as image segmentation. The image can be a grayscale or color image.

## 7. REFERENCES

- [1] Eberhart, R. and Kennedy, J. 1995. Particle swarm optimization, *IEEE Int. Conf. Neural Networks*, pp. 1942–8.
- [2] Shi, E. and Eberhart, R. 1998. Modified particle swarm optimizer, *Proc. IEEE Conf. Evol. Comput. ICEC*, pp. 69–73.
- [3] Eberhart and Yuhui Shi. 2002. Particle swarm optimization: developments, applications and resources,” no. February 2001, pp. 81–86.
- [4] Yong, D. Chuansheng, W. and Haimin, G. 2018. Particle Swarm Optimization Algorithm with Adaptive Chaos Perturbation, *Cybern. Inf. Technol.*, vol. 15, no. 6, pp. 70–80.
- [5] Dileep, M. V. and Kamath, S. 2015. A review on particle swarm optimization algorithm and its developments,” *Glob. J. Pure Appl. Math.*, vol. 11, no. 6, pp. 4997–5018.
- [6] Mengxia, L. Ruiquan, L. and Yong, D. 2016. The Particle Swarm Optimization Algorithm with Adaptive Chaos Perturbation,” vol. 11, no. December, pp. 804–818.
- [7] Xin, J. Chen, G. and Hai, Y. 2009. A particle swarm optimizer with multi-stage linearly-decreasing inertia weight,” *Proc. 2009 Int. Jt. Conf. Comput. Sci. Optim. CSO 2009*, vol. 1, pp. 505–508, 2009, doi: 10.1109/CSO.2009.420.
- [8] Khare, A. and Rangnekar, S. 2013. A review of particle swarm optimization and its applications in Solar Photovoltaic system,” *Appl. Soft Comput. J.*, vol. 13, no. 5, pp. 2997–3006.
- [9] Feng, Y. Teng, G. F. Wang, A. X. and Yao, Y. M. 2008. Chaotic inertia weight in particle swarm optimization, *Second Int. Conf. Innov. Comput. Inf. Control. ICICIC 2007*, pp. 7–10.
- [10] Malik, R. Rahman, F. T. a . Hashim, S. Z. M. and Ngah, R. 2007. New Particle Swarm Optimizer with Sigmoid Increasing Inertia Weight,” *Int. J. Comput. Sci. Secur. IJCSS*, vol. 1, no. 1, pp. 35–44.
- [11] Kentzoglanakis, K. and Poole, M. 2009. Particle swarm optimization with an oscillating inertia weight,” *Proc. 11th Annu. Genet. Evol. Comput. Conf. GECCO-2009*, no. January 2009, pp. 1749–1750.
- [12] Gao, Y. L. An, X. H. and Liu, J. M. 2008. A particle swarm optimization algorithm with logarithm decreasing inertia weight and chaos mutation,” *Proc. - 2008 Int. Conf. Comput. Intell. Secur. CIS 2008*, vol. 1, pp. 61–65, 2008.
- [13] Li, H. R. and Gao, Y. L. 2009. Particle swarm optimization algorithm with exponent decreasing inertia weight and stochastic mutation,” *2009 2nd Int. Conf. Inf. Comput. Sci. ICIC 2009*, vol. 1, no. 3, pp. 66–69.
- [14] Borowska, B. 2017. Exponential inertia weight in particle swarm optimization,” *Adv. Intell. Syst. Comput.*, vol. 524, no. January 2015, pp. 265–275, 2017.
- [15] Cui, Q. 2017. Globally-optimal prediction-based adaptive mutation particle swarm optimization,” *Inf. Sci. (Ny).*, vol. 418–419, pp. 186–217.