New Particle Swarm Optimizer with Sigmoid Increasing Inertia Weight

Reza Firsandaya Malik reza_malik2000@yahoo.com
Wireless Communication Centre
Faculty of Electrical Engineering
University Technology Malaysia
Johor Bahru, 81310, Malaysia

Tharek Abdul Rahman tharek@fke.utm.my
Wireless Communication Centre
Faculty of Electrical Engineering
University Technology Malaysia
Johor Bahru, 81310, Malaysia

Siti Zaiton Mohd. Hashim sitizaiton@utm.my
Faculty of Computer Science and Information System
University Technology Malaysia
Johor Bahru, 81310, Malaysia

Razali Ngah razalin@fke.utm.my
Wireless Communication Centre
Faculty of Electrical Engineering
University Technology Malaysia
Johor Bahru, 81310, Malaysia

Abstract

The inertia weight of particle swarm optimization (PSO) is a mechanism to control the exploration and exploitation abilities of the swarm and as mechanism to eliminate the need for velocity clamping. The present paper proposes a new PSO optimizer with sigmoid increasing inertia weight. Four standard non-linear benchmark functions are used to confirm its validity. The comparison has been simulated with sigmoid decreasing and linearly increasing inertia weight. From experiments, it shows that PSO with increasing inertia weight gives better performance with quick convergence capability and aggressive movement narrowing towards the solution region.

Keywords: Particle Swarm Optimization, Inertia Weight, Linearly Increasing Inertia Weight, Sigmoid Decreasing Inertia Weight, Sigmoid Increasing Inertia Weight.

1. Introduction

Particle Swarm Optimization (PSO) is population based stochastic optimization technique inspired by social behavior of bird flocking and fish schooling [1]. The PSO algorithm was first introduced by Erberhart and Kennedy in 1995 [1, 2]. A PSO algorithm maintains a swarm of particles, where each represents a potential solution. In analogy with evolutionary computation paradigms, a swarm is similar to a population, while a particle is similar to an individual. Each particle adjusts its trajectory towards the best its previous position attained by any member of its neighborhood or globally, the whole swarm. The particles are flown through multidimensional search space, where the position of each particle adjusted according to its own experience and that of its neighbors. The movement of each particle in search space with adaptive velocity and store the best position of the search space it has ever visited. The particles search for best position until a relatively unchanging state has been encountered or until computational limitation exceeded.

Since its introduction, PSO has seen many improvements and applications. Most modifications to the basic PSO are directed towards improving convergence of the PSO and increasing the diversity of the swarm [3]. The modification in PSO consists of three categories: extension of field searching space [4], adjustment the parameters [5], and hybrid with another technique [6]. A number of parameters modification include inertia weight, velocity clamping, velocity constriction, cognitive and social coefficient, different ways of determining the personal best (*pbest*) and global best (*gbest*) positions, and different velocity models. The modification of basic PSO was reported in [7 - 9] that introduced new methods of inertia weight which tuned based on trial and error. Suitable selection of the inertia weight provides a balance between global and local searching. In these concepts proposed a linearly decreasing, linearly increasing and sigmoid decreasing inertia weight to get better PSO performance. There are advantages between three methods which is sigmoid decreasing inertia has near optimum solution better than the others and linearly increasing weight has quick convergence ability better than the others. For Linear decreasing has near optimum solution better than linear increasing inertia weight (LIIW).

The efficiency of PSO is expressed as the number of iterations or generations to find optimum solution with specified accuracy. With less generation, the near optimum solution can be reach with quick convergence ability from swarm. This paper presents alternative solution for quick convergence and maximum near optimum solution. The method will be combination between sigmoid decreasing and linear increasing to fulfill the objective of this paper. The method of sigmoid increasing inertia weight (SIIW) will have quick convergence ability and aggressive movement narrowing down towards the solution region. The schema attempted to increase inertia weight by means of sigmoid function. In this work some empirical studies are investigated. In Section 2, philosophy and procedure of original PSO are explained and then the standard PSO with a decreasing and increasing inertia weight and sigmoid decreasing inertia weight (SDIW) in short presented. In Section 3, a new PSO model with a sigmoid increasing inertia weight is suggested. To prove the validity of such methods, several standard benchmark functions are tested in Section 4. The empirical data resulted will be emphasized and discussed in Section 5. Finally Section 6 concludes this paper.

2. PSO Algorithm

2.1 Simple PSO

The PSO concept consists of changing the velocity each particle toward its *pbest* and *gbest* positions at each time step. Velocity is weighted by a random term, with separate random numbers generated for velocity toward *pbest* and *gbest* positions. The process of PSO can be described as follows:

1. Initialize a population (array) of particles with random positions and velocities on d dimensions in the problem space.
2. For each particle, evaluate the desired optimization fitness function in d variables.
3. Compare particle's fitness evaluation with particle's *pbest*. If current value is better than pbest, then set pbest position equal to current position in D dimensional space.
4. Compare fitness evaluation with the population's overall previous best. If current value is better than *gbest*, then reset *gbest* to the current particle's array index and value.
5. Change the velocity and position of the particle according to equations (1) and (2) respectively:

$$v_i^{k+1} = v_i^k + c_1 * rand(.) * (pbest - x_i^k) + c_2 * rand(.) * (gbest - x_i^k) \quad (1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{2}$$

where v_i^k, v_i^{k+1}, and x_i^k are velocity vector, modified velocity and positioning vector of particle i at generation k, respectively. The c_1 and c_2 are cognitive and social coefficients that influence particle velocity.

6. Loop to step 2 until a criterion is met, usually a sufficiently good fitness or a maximum number of iterations (generations).

The maximum velocity *Vmax* serves as a constraint to control the global exploration ability of a particle swarm. Exploration is the ability to test various regions in the problem space in order to locate a good optimum. If *Vmax* is too high particles might fly past good solutions and facilitate global exploration. If *Vmax* is too small particles may not explore sufficiently beyond locally good regions and encourage local exploitation. Exploitation is the ability to concentrate the search around a promising candidate solution in order to locate the optimum precisely. When local exploitation happens, they could trap in local optima, unable to move far enough to reach a better position in the problem space.

Generally, balancing between exploration and exploitation searching process will improve PSO performance. This exploration and exploitation tradeoff is influenced by modifying and tuning some parameter, namely current motion, inertia weight, cognitive and social coefficients.

2.2 Inertia Weight

The concept of an inertia weight was developed to better control exploration and exploitation. The aim of inertia weight was to be able to control the exploration and exploitation mechanism and eliminate the need for *Vmax*. The inclusion of an inertia weight in the PSO algorithm was first published in 1998 [5]. The inertia weight was successful in addressing first aim but could not completely eliminate the need of velocity clamping. The inertia weight (*w*) controls the momentum of the particle by weighting the contribution of the previous velocity. Equation (3) and (4) describe the velocity and position update equations with an inertia weight included. It can be seen that these equations are identical to equations (1) and (2) with the addition of the inertia weight w as a multiplying factor of v_i^k in equation (3).

$$v_i^{k+1} = w * v_i^k + c_1 * rand(.) * (pbest - x_i^k) + c_2 * rand(.) * (gbest - x_i^k) \tag{3}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{4}$$

In previous works, implementation of the inertia weight used a constant [5] and dynamic [7 – 12] value for entire search duration and for all particles for each dimension. For constant value, the velocity constants cognitive (c_1) and social (c_2) coefficient in equation (3) represent the weighting of the velocity. There are two different approaches for dynamic value which is decreasing and increasing. For decreasing, an initially large value of inertia weight decrease linearly or nonlinearly to a small value. A large inertia weight facilitates a global search while a small inertia weight facilitates a local search. For increasing, a small inertia weight increase linearly or nonlinearly to a larger value in linearly increasing. A large inertia weight has more possibility to converge, which implicates a larger inertia weight in the end of search will foster the convergence ability. In [5, 7], There are many methods in nonlinear approaches such as sigmoid function [9], tracking and dynamic system [11] and constriction factor [12]. Shi suggested that an inertia weight value starting from 0.9 linearly decreasing to 0.4. In [8], Y. Zheng, et. al. suggested that an inertia weight value starting from 0.4 linearly increasing to 0.4. The value proposed by Y. Zheng, et. al. adopted in experiment to give the PSO a better performance.

3. PSO with Sigmoid Increasing Inertia Weight

This work proposes a new inertia weight modulated with sigmoid function for improving the performance of PSO. Based on the detail observation and analysis, this work has been inspired by the excellence performance show by linearly increasing and sigmoid decreasing inertia weight which one state and discuss a little bit hence providing sigmoid increasing inertia weight (SIIW) approach. The forms of sigmoid function either in the form of sigmoid decreasing and sigmoid increasing are present in figure 1.

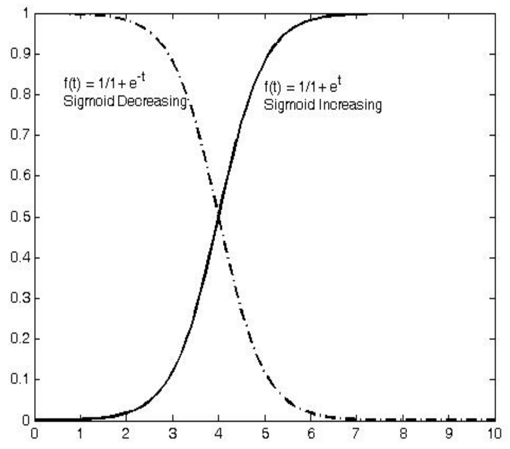


FIGURE 1: Sigmoid Decreasing and Increasing Inertia Weight

The basic of sigmoid function is given as:

$$f(t) = \frac{1}{1+e^{-t}} \qquad (5)$$

The equation (5) utilized in equation (6) used by SDIW and SIIW in equation (7) as following equations:

$$w_k = \frac{(w_{start} - w_{end})}{(1+e^{-u*(k-n*gen)})} + w_{end} \qquad (6)$$

$$w_k = \frac{(w_{start} - w_{end})}{(1+e^{u*(k-n*gen)})} + w_{end} \qquad (7)$$

$$u = 10^{(\log(gen)-2)} \qquad (8)$$

where:
w_k is inertia weight at k, w_{start} and w_{end} are inertia weight at the start and inertia weight at the end of a given run, respectively. Furthermore, u is the constant to adjust sharpness of the function, *gen* is the maximum number of generations to run and n is the constant to set partition of sigmoid function.

The figure (5) utilized in equation (6) based on PSO process. Using the equation (6), the inertia weight will implement in sigmoid curve. In [9], the sigmoid curve as shown in figure 1 for equation (6) as known as sigmoid decreasing inertia weight. The sigmoid shape in sigmoid decreasing inertia weight same with the basic sigmoid function. In sigmoid decreasing inertia weight, a large inertia weight is maintained in first part of PSO process to assure the global search. Afterwards, a small inertia weight is retained to facilitate a local search in final part of PSO process.

The equation (7) is opposite from equation (6) and known as sigmoid decreasing. In sigmoid increasing weight, a small inertia weight is maintained in first part of PSO process to local search. This process to beginning facilitate the PSO to avoid been attracted to local optima, explore the whole solution space and makes out the correct direction [7]. Afterwards, a large inertia weight is retained to facilitate global optima more efficiently in the end of PSO process. There is gradation between small and large value for local and global search. However, such alteration improves the quick convergence ability and maximum optimum solution prominently. The experiment results are shown in the next section.

4. Validating New PSO Optimizer

For comparison, four non-linear functions used in [7] are used as benchmark functions for observing the performance of the proposed optimizer, compared to others. The main objectives are to achieve faster convergence ability and near optimum solution, the experiment results were presented in graphs and tables.

The first function is the Sphere function described by equation (9):

$$f_0(x) = \sum_{i=1}^{n} x_i^2 \tag{9}$$

where $x = [x_1, x_2, \ldots, x_n]$ is an n-dimensional real-valued vector. Then, the second function is the Rosenbrock function given as (10):

$$f_1(x) = \sum_{i=1}^{n} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2) \tag{10}$$

The third function is the generalized Rastrigrin function described by equation (11):

$$f_2(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10) \tag{11}$$

The last function is the generalized Griewank function described by equation (12):

$$f_3(x) = \frac{1}{400}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1 \tag{12}$$

For the purpose of evaluation, the asymmetric initialization method [6] was adopted here for population initialization. Table 1 lists the initialization ranges of the four functions:

Function	Asymmetric Initialization Range
f_0	$(50 , 100)^n$
f_1	$(15 , 30)^n$
f_2	$(2.56 , 5.12)^n$
f_3	$(300 , 600)^n$

TABLE 1: Asymmetric initialization ranges.

For each function, three different dimension sizes are tested. They are population sizes: 20, 40 and 80. The maximum number of generations is set as 1000 and 2000 corresponding to the dimensions 20, 40 and 80, respectively. In order to investigate whether the PSO algorithm scales well or not, different population sizes are used for each function with different dimensions. A sigmoid decreasing or increasing inertia weight is used at 0.4 until 0.9, which $c_1 = 2$ and $c_2 = 2$. When objective to find the best partition of sigmoid function, different sigmoid constants, n, are used; they are 0.25, 0.5 and 0.75. The best function value (minimum) will be observed in these experiments.

5. Results and Discussions

As the objective of searching methods was to achieve faster convergence ability and aggressive movement narrowing down towards the solution region, the experiment results will be shown in table and graphs. Linear Decreasing Inertia Weight (LDIW), Linear Increasing Inertia Weight (LIIW) and Sigmoid Decreasing Inertia Weight (SDIW) used as comparison to proposed method.

In all figures show that x-axis and y-axis represent number of generation (iteration) and value of calculation fitness function, which characterizes the optimization problem.

Figure 2 shows the results for the Sphere function, figure 3 the Rosenbrock function, figure 4 the Rastrigrin function, and figure 5 the Griewank function with three different population sizes, respectively. In figure 2 shows that, SIIW and LIIW have quick convergence better than SDIW. This convergence ability for both methods to reach minimum fitness function has been achieved at 200th generation. For SDIW, the minimum fitness function can be achieve start at 600th (n = 0.25), 1,000th (n=0.5) and 1,600th (n=0.75) generation.

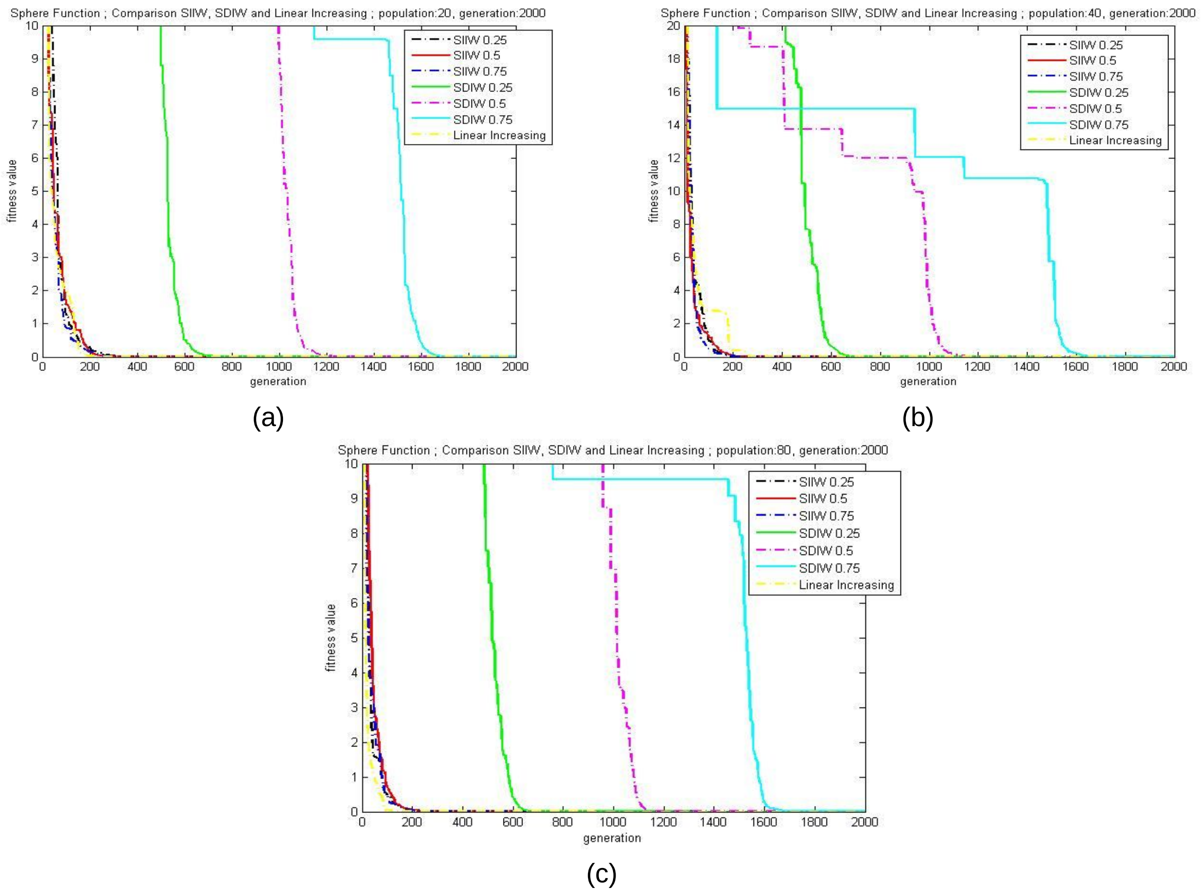


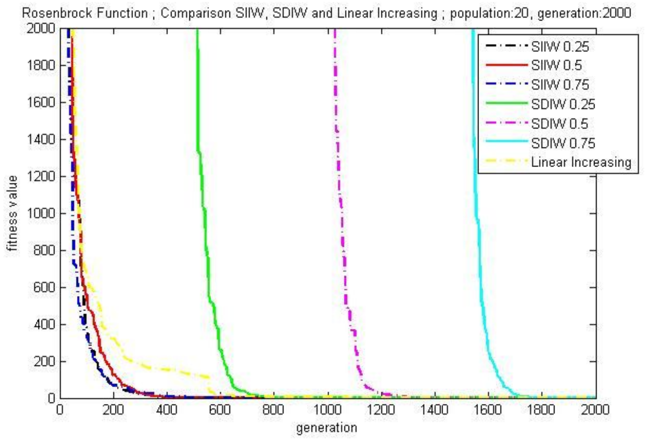
FIGURE 2: Curve of Sphere Function with population size: (a) 20, (b) 40, and (c) 80.

Pop Size	Gen	Mean Best Function Value						
		SIIW			SDIW			Linear Increasing
		0.25	0.5	0.75	0.25	0.5	0.75	
20	1000	1.9549E-07	3.6991E-07	2.2936E-09	5.9805E-11	2.7347E-07	2.6000E-03	1.0293E+00
	2000	3.9212E-10	2.4862E-14	3.4405E-17	3.5240E-20	1.9234E-13	7.3342E-07	9.0548E-06
40	1000	6.6276E-12	6.2689E-10	1.7541E-11	2.9253E-13	3.6077E-09	3.6963E-04	1.1940E-01
	2000	3.2614E-19	2.0522E-22	1.4664E-24	6.4662E-27	1.3518E-17	2.5005E-08	6.8401E-18
80	1000	7.0224E-14	5.9036E-14	1.4806E-14	1.5966E-15	1.7712E-10	2.2310E-05	1.2780E-15
	2000	3.7240E-30	3.0725E-30	5.5822E-29	1.0025E-30	3.6792E-20	5.3844E-10	3.8843E-29

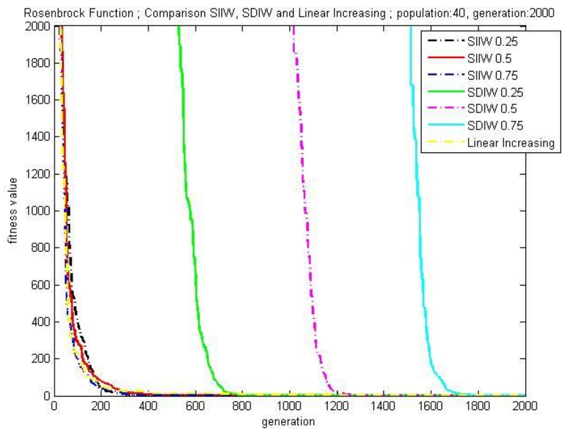
TABLE 2: The mean fitness values for the Sphere function.

Table 2 shows that, the SIIW and SDIW share some similar convergent characteristic with each other. With a given population size, SIIW has the best value at n = 0.75 (5.5822E-29) and SDIW at n = 0.25 (1.0025E-30). The SIIW, SDIW and LIIW find more precise result on large population size.

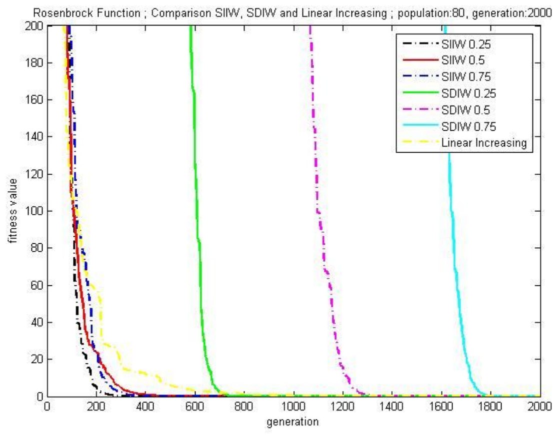
In figure 3 shows that, SIIW and LIIW have quick convergence better than SDIW. This convergence ability for both methods to reach minimum fitness function has been achieved at 200th generation. For SDIW, the minimum fitness function can be achieve start at 600th (n = 0.25), 1,000th (n=0.5) and 1,600th (n=0.75) generation.



(a)



(b)



(c)

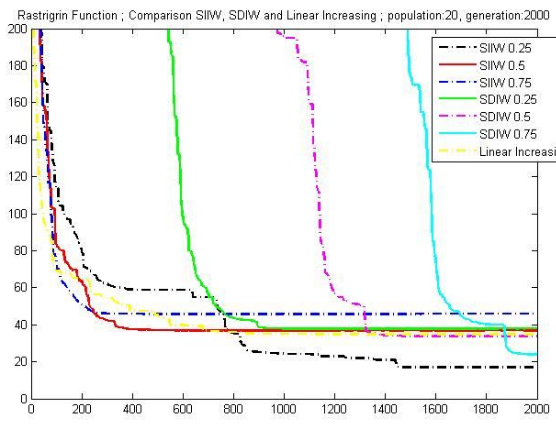
FIGURE 3: Curve of Rosenbrock Function with population size: (a) 20, (b) 40, and (c) 80.

Pop Size	Gen	Mean Best Function Value						
		SIIW			SDIW			Linear Increasing
		0.25	0.5	0.75	0.25	0.5	0.75	
20	1000	0.7048	0.1490	0.0100	0.0010	0.0742	5.1095	1.7283
	2000	0.0438	0.0384	0.0264	0.0023	0.0002	0.2413	1.7419
40	1000	0.2463	0.0410	0.0223	0.0200	0.0143	0.7881	0.4300
	2000	0.0221	0.0028	0.0478	0.0038	0.0043	0.0360	0.2362
80	1000	0.0969	1.4994E-04	0.0286	0.0091	0.0480	0.1654	0.0164
	2000	0.0408	0.0154	0.0062	0.0041	0.0067	0.0026	0.1064

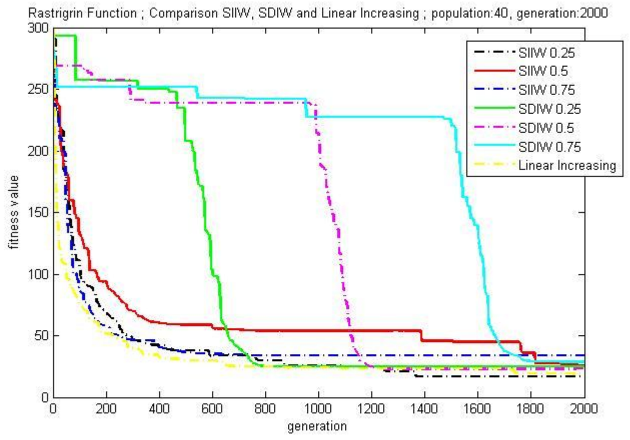
TABLE 3: The mean fitness values for the Rosenbrock function.

Table 3 shows that, the SDIW (n=0.25, 0.5) has better minimum function value than others. But for the best value in the Rosenbrock function is SIIW at n = 0.5 (1.4994E-04).

In figure 4 shows that, SIIW and LIIW have quick convergence better than SDIW. The SIIW has better minimum function better than others. This convergence ability for all methods can not be reach the lowest minimum fitness function compare with the result from Sphere and Rosenbrock functions.

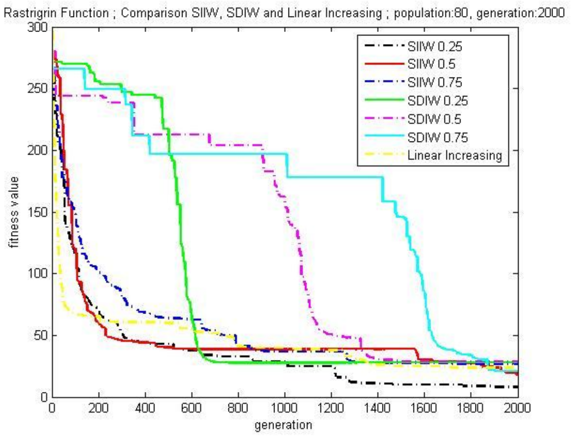


(a)



(b)

(c)



(c)

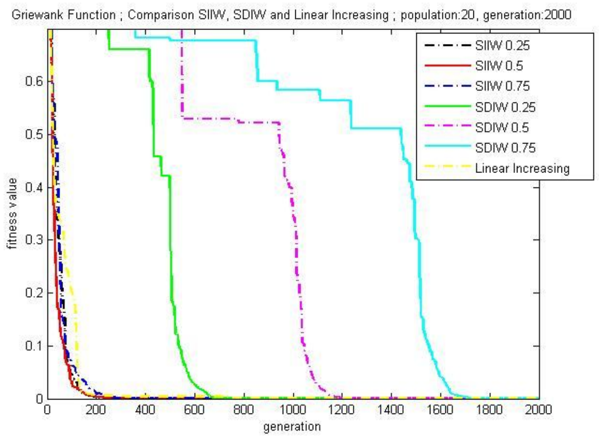
FIGURE 4: Curve of Rastrigrin Function with population size: (a) 20, (b) 40, and (c) 80.

Pop Size	Gen	Mean Best Function Value						
		SIIW			SDIW			Linear Increasing
		0.25	0.5	0.75	0.25	0.5	0.75	
20	1000	34.7488	33.1415	35.8835	37.8084	34.9509	48.4652	39.0512
	2000	16.9194	36.8135	45.7681	37.8084	33.8286	23.9116	34.8263
40	1000	24.0228	29.8547	24.8773	38.8034	30.8787	39.6623	34.8307
	2000	16.9143	25.8745	33.8286	24.8739	22.8840	28.8540	19.0803
80	1000	13.9377	21.9866	30.8449	31.8387	29.8509	33.5563	32.8748
	2000	7.9599	17.4373	26.8639	27.8588	28.8538	20.8963	23.8791

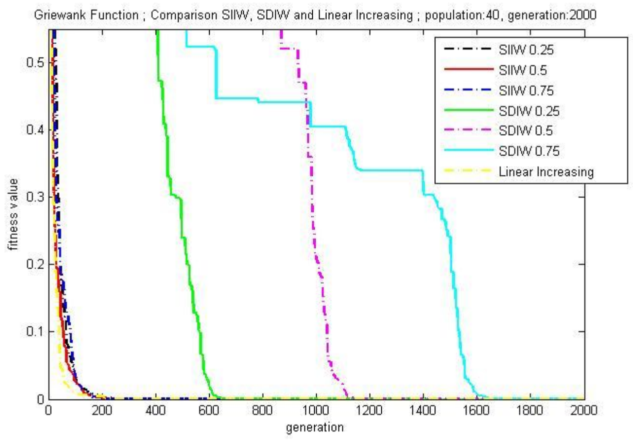
TABLE 4: The mean fitness values for the Rastrigrin function.

Table 4 shows that, the best value for the Rastrigrin in any scenarios is SIIW at n = 0.25 (7.9599).

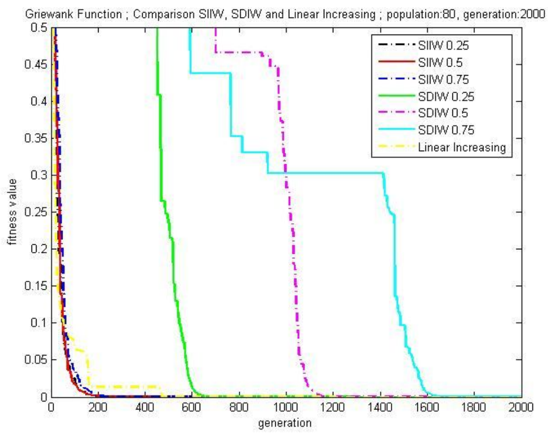
In figure 5 has similarity conclusion with figures 2 and 3. The SIIW still the best quick convergence ability in these figures.



(a)



(b)



(c)

FIGURE 5: Curve of Griewank Function with population size: (a) 20, (b) 40, and (c) 80.

Pop Size	Gen	Mean Best Function Value						
		SIIW			SDIW			Linear Increasing
		0.25	0.5	0.75	0.25	0.5	0.75	
20	1000	1.1265E-07	1.4556E-09	1.9950E-10	1.4892E-11	4.9864E-08	6.1981E-05	1.0100E-02
	2000	4.0649E-12	4.4409E-16	0	0	4.3299E-15	1.6194E-07	1.2000E-03
40	1000	1.3895E-11	3.0220E-12	3.2052E-13	2.4425E-14	2.5046E-10	6.2352E-06	1.9661E-11
	2000	0	0	0	0	0	5.5402E-10	6.9356E-13
80	1000	1.1102E-16	6.6613E-16	3.3307E-16	0	9.9489E-12	1.6527E-06	0
	2000	0	0	0	0	0	2.2026E-11	0

TABLE 5: The mean fitness values for the Griewank function.

Table 5 shows that, the third PSO can reach minimum fitness function (0) but the SIIW and LIIW still has quick convergence ability better than SDIW (see figure 5). By looking at the shape of the curves in all figures, PSO with SIIW and LIIW converge quickly under any cases. Compare to LIIW, the proposed PSO can be improved greatly and have better result. Tables and figures above all indicate that the new proposed method (SIIW) has improvement when increasing population size, generation and sigmoid constant (*n*).

6. Conclusion and Future Work

In this paper, the performance of the PSO algorithm with sigmoid increasing inertia weight has been investigated and extensively. The results are compared with the PSO with sigmoid decreasing and linearly inertia weight by experimenting on four non-linear benchmark functions well studied in the literature. The sigmoid function has contributed to getting minimum fitness function while linearly increasing inertia weight give contribution to quick convergence ability. The combination of sigmoid function and linear increasing inertia weight in SIIW has produced a great improvement in quick convergence ability and aggressive movement narrowing towards the solution region with different sigmoid constant (*n*). From the experiment results, it shows that the sigmoid constant (*n*) plays an important role in searching for optimal solution in SIIW.

For future work, the proposed SIIW will be implemented and tested in real application to further verify the results.

7. References

1. J. Kennedy, and R.C. Eberhart, "*Particle swarm optimization*", Proceedings of the 4th IEEE International Conference on Neural Networks, pp. 1942 – 1948, 1995.
2. R.C. Eberhart, and J. Kennedy, "*A new optimizer using particle swarm theory*", Proceedings of 6th International Symposium on Micro Machine and Human Science, pp. 39 – 43, 1995.
3. A. P. Engelbrecht, "*Fundamentals of Computational Swarm Intelligence*", John Willey & Sons Inc., pp. 93 (2005)
4. J. Kennedy, and R.C. Erberhart, "*A discrete binary version of the particle swarm algorithm*", Proceeding of 1997 Conference on Systems, Man, and Cybernetics, Florida, USA, 1997.
5. Y. Shi, and R.C. Erberhart, "*Parameter selection in particle swarm optimization*", Proceedings of 7th Annual Conference on Evolution Computation, pp. 591 – 601, 1998.
6. M. Lovbjerg, T.K. Rasmussen, and T. Krink, "*Hybrid particle swarm optimizer with breeding and subpopulation*", Proceeding of the Third Genetic and Evolutionary Computation Congress, pp. 469 – 476, San Fransisco, CA, 2001.
7. Y. Shi, and R.C. Eberhart, "*Empirical study of particle swarm optimization*", Proceedings of the 1999 Congress on Evolutionary Computation, pp. 1945 – 1950, 1999.
8. Y. Zheng, et. al., "*Empirical study of particle swarm optimizer with an increasing inertia weight*", Proceeding of the IEEE Congress on Evolutionary Computation, 2003.
9. A. Adriansyah, and S.H.M. Amin, "*Analytical and empirical study of particle swarm optimization with a sigmoid decreasing inertia weight*", Regional Conference on Engineering and Science, Johor, 2006
10. A. Chatterjee, and P. Siarry, "*Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization*", Computer and Operation Research, 33:859 – 871, 2006.
11. R.C. Eberhart, and Y. Shi, "*Tracking and optimizing systems with particle swarms*", Proceeding Congress on Evolutionary Computation, 2001, Seoul, South Korea.
12. M. Clerc, "*The swarm and the queen: towards a deterministic and adaptive particle swarm optimization*", Proceeding of the IEEE Congress on Evolutionary Computation, pp. 1951 – 1957, Washington D.C, 1999.