IMPLEMENTATION OF PSO AND FIPSO WITH ONSIDERATION OF CONSTANT AND LINEARLY DECREASING WEIGHT STRATEGIES ON MICHAELWICZ 3-D FUNCTION

Muhammad Salman Fakhar[1], Syed Abdul Rahman Kashif[2], Hafiz Zaheer Hussain[3], Bilal Ashfaq Ahmad[4]
[1,3,4] The University of Lahore
[2] University of Engineering and Technology Lahore
Email:salman.fakhar@ee.uol.edu.pk, abdulrahman@uet.edu.pk, zaheer.hussain@ee.uol.edu.pk, bilal.ashfaq@ee.uol.edu.pk

ABSTRACT*-Particle swarm optimization (PSO) has gained a lot popularity among heuristic optimization algorithms for it helps in achieving near to global optimal solution of different objective functions. Similar is the case with its variants. To test the power of any optimization algorithm, test functions, mostly multimodal in nature are usually deployed, and the algorithms are implemented on them to find their global optima. Michaelwicz 3D function is one such type of multi modal function. This paper presents the implementation of PSO and its variant FIPSO, while considering two types of weight strategies i.e. constant weight and linearly decreasing weight, on this function.*

Index Terms: PSO, FIPSO, weight, optimization, Michaelwics 3D function.

1. INTRODUCTION

PSO has now become a renowned meta-heuristic algorithm which was inspired from the social behaviors of the fish and birds while they search for food or find new habitats according to meet their needs. It was first presented by Kennedy and Eberhart in 1995. Though, the other famous meta-heuristic algorithms like genetic algorithm also follow the similar intelligence. Yet particle swarm optimization has gained fame for its ease in implementation for it uses the haphazardness of real numbers [1-2]. Since the birth of Particle Swarm Optimization algorithm, many variants of it have been coming into existence for it is believed that the canonical version is sometimes not able to reach the global optima of the objective function. A new extension to the canonical PSO was recently introduced by the author of the first version, i.e. Kennedy along with Mendes and Neves as given in reference [1-2]. However, reference [2] also suggested that FIPSO implementations are not overtaking PSO implementations on all types of test functions.

To reveal and explain the Original Particle Swarm Optimization and the FIPSO algorithm, a test objective function is taken which is of two dimensions and is optimized by using the very algorithm. It is a multimodal function i.e. a function having numerous peaks. The function has been taken from source [3].

2. MICHAELWICZ 3D FUNCTION

The 2 dimensional Michaelewicz function, is given as

$$f(\mathrm{x,y}) = -\left\{ \sin(x)\left[\sin\left(\frac{x^2}{\pi}\right)\right]^{2m} + \sin(y)\left[\sin\left(\frac{2y^2}{\pi}\right)\right]^{2m} \right\} \quad (1)$$

Where,

m = 10 in this paper. According to reference [4], the value of "m" changes the function in terms of the number of minima it has. However, to perform the test in this paper, the value of "m" has been taken as 10. The Michaelwicz 3D function is shown in Figure 1.

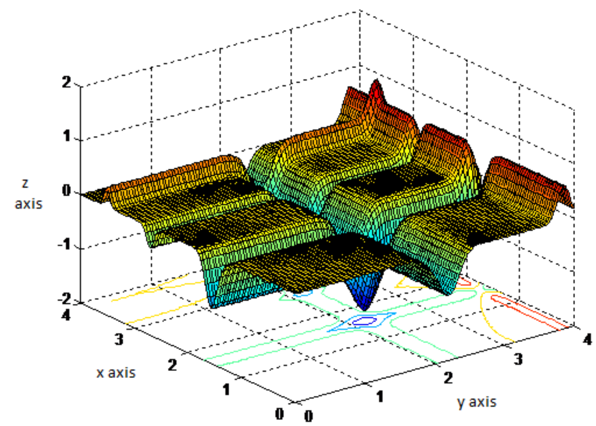

Figure 1: Michaelwicz 3D function with m = 10

It can be seen in Figure 1 that the graph has multiple minima, however its global or ultimate minima is only one.

3. PSO AND FIPSO IMPLEMENTATIONS

The canonical PSO iterations are proceeded as;

$$V_{i+1} = R\left(V_i + Rand\left(0,\frac{\varnothing}{2}\right).(P_i - X_i) + Rand\left(0,\frac{\varnothing}{2}\right).(P_g - X_i)\right) \quad (2)$$

$$X_{i+1} = X_i + V_{i+1} \quad (3)$$

Where,

X_i is the existing position of the particle, X_{i+1} is the new position of the particle V_i is the velocity to be added in previous location of particle to proceed forward R is the restriction coefficient or also known as weight according to reference [2]. U is the random vector generator function. P_i is the local best of particle and P_g is the global best of particles.

In Fully Informed PSO, each individual particle is fully informed with its neighborhood. Its iterations are given as:

$$\vartheta_{i+1} = R\left(\vartheta_i + \frac{\sum_{n=1}^{N_i} Rand\,(0,\phi).(P_{nbr(n)} - X_i)}{N_i} \right) \quad (4)$$

$$X_{i+1} = X_i + \vartheta_{i+1} \quad (5)$$

Where,

$P_{nbr(n)}$ is the neighbor of particle X*i*. *R* is the restriction coefficient also known as weight in text. ϑ_i is the velocity of particle *i* as mentioned in reference [2].

According to reference [2], FIPSO should have the tendency of searching the complete search space, however, the implementations showed that FIPSO was not a suitable algorithm for most of the test functions. In this paper FIPSO and PSO will be shown implemented on the Michaelwicz 3D function.

In this paper, both PSO and FIPSO are tested on Michaelwicz 3D function while considering two types of weights or restriction coefficients as mentioned in reference [5]. In both implementations, R or weight can be taken as constant, usually taken in texts as 0.729. Also, on both the implementations, the weight is considered to linearly decrease from 0.3 to 0.1 by using the equation (6).

$$W = W_{\max} - \left(\frac{W_{\max} - W_{\min}}{iteration_{\max}} \right) \times iteration_{present} \quad (6)$$

3. RESULTS

Table 1 gives the results of the implementations of PSO and FIPSO, using the two mentioned weight strategies, on the Michelwicz 3D function. Another result of the implementation of accelerated PSO, yet another variant of PSO, as given in reference [3], is also mentioned in table 1. The results in table 1 do not present a completely true story as it gives the results of the final iterations of the best performances of all these variants of PSO. To make a consistent analysis of the performances of all these algorithms, the number of particles are taken as 8. The neighborhood topology considered is the *gbest* neighborhood topology as mentioned in references [1-2]. The number of iterations performed is 50 for all these variants. However, reference [3] has performed only 15 iterations for the implementation of accelerated PSO on the Michelwicz equation. The results of the implementations are also shown in the form of contouring of particles with all the four implementations. In figures 2 to 5.

In figures 2 to 5, the places of particles in the last iteration are shown on the contour charts. It can be seen both on the contour and the statistical charts of figures 5 to 8 also strengthen the story that PSO is performing better with both the weight strategies as compared to FIPSO with both the weight strategies. FISPO is rather not able to find the near approximation to global minimum for the 100 runs each with both the weight strategies.

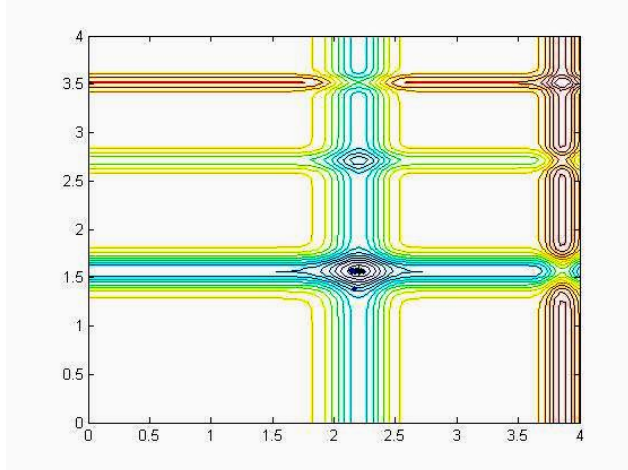

Figure 2: Contour of particles on last iteration using PSO with constant weight technique

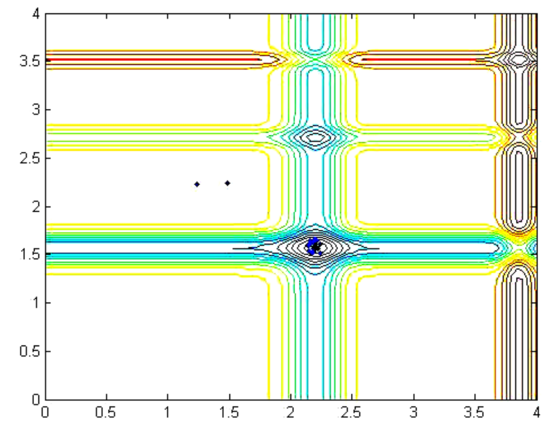

Figure 3: Contour of particles on last iteration using PSO with linearly decreasing weight technique

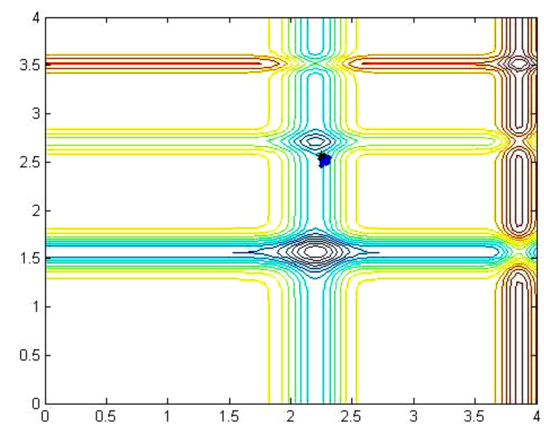

Figure 4: Contour of particles on last iteration using FIPSO with constant weight technique

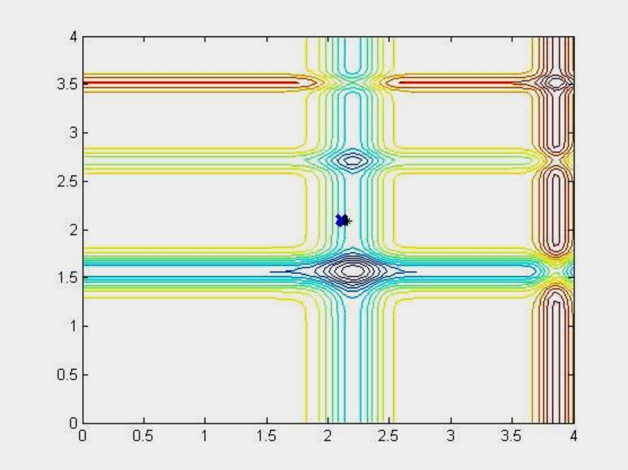

Figure 5: Contour of particles on last iteration using FIPSO with linearly decreasing weight technique

Table 1: Best results of implementation of PSO variants using two weight strategies

ALGORITHM	X	Y	Z
PSO WITH CONSTANT WEIGHT	**2.2038**	**1.5765**	**-1.8013**
PSO WITH LINEARLY DECREASING WEIGHT	**2.1977**	**1.5711**	**-1.8009**
FIPSO WITH CONSTANT WEIGHT	**2.2675**	**1.6328**	**-1.5845**
FIPSO WITH LINEARLY DECREASING WEIGHT	**2.0653**	**1.5743**	**-1.5567**
ACCELERATED PSO [3]	**2.20319**	**1.57049**	**-1.801**

To know which variants of PSO have performed best on Michaelwicz 3D function, statistical data of the implementations of each of the four types is also given in the form of bar charts for 100 runs of each variant. This is given in figures 4 to 8.

The statistical data presented in figures 6 to 9 is mentioning that though FIPSO could have been the best variant in terms of its optimum searching capability, however, it has astonished by not giving the minimum results for both the weigh strategies. PSO with its both variants have proved itself a better option for test functions like Michaelwicz 3D function. FIPSO gives the solution for a very less number of times as compared to PSO and accelerated PSO.

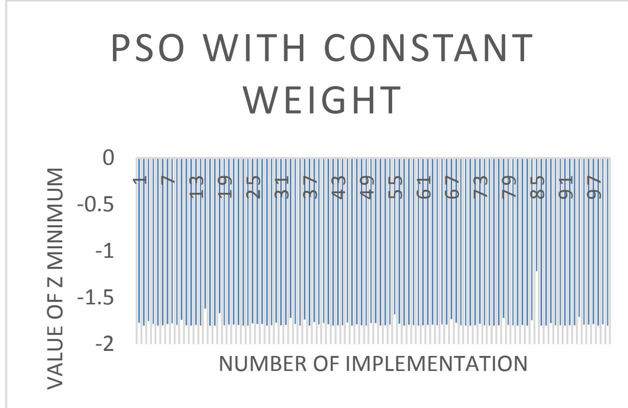

Figure 6: Statistical data of PSO with constant weight technique

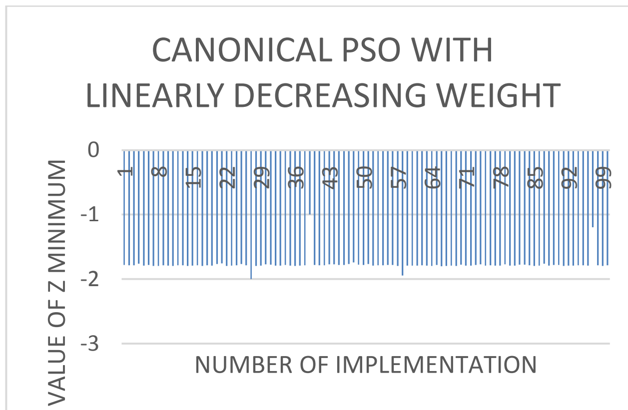

Figure 7: Statistical data of PSO with linearly decreasing weight technique

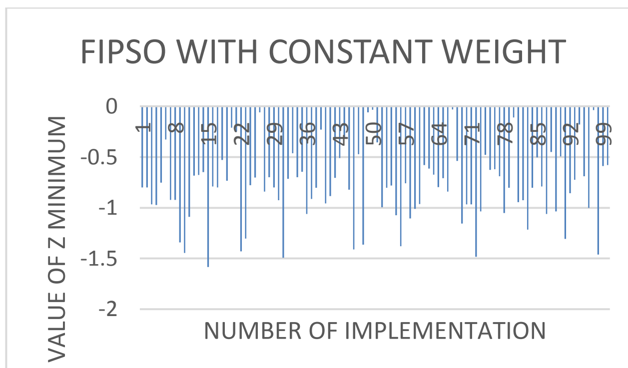

Figure 8: Statistical data of FIPSO with constant weight technique

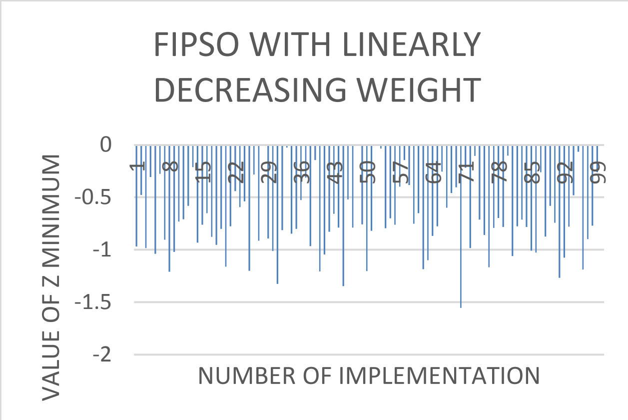

Figure 9: Statistical data of FIPSO with linearly decreasing weight technique

FIPSO has been observed to converge its particles but sticks to the local minima. May be this is because each particle in FIPSO has more information than the particles in PSO and accelerated PSO.

6. CONCLUSION

Implementation of PSO and FIPSO and its comparison with accelerated PSO has shown that FIPSO response towards multimodal test functions like Michaelwicz 3D function is not that good as PSO and accelerated PSO have on the very function. FIPSO usually sticks to the local minima while PSO and accelerated PSO move towards global minimum and most of the times the particles converge towards the global minimum in PSO and accelerated PSO implementations. The time to converge to the solution is less than a minute in all these implementations in most of the modern computers.

REFERENCES

[1] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler, maybe better", *IEEE Transactions on Evolutionary Computation*,**vol. 8**, pp. 204-210, 2004.

[2] J. Kennedy and R. Mendes, "Neighborhood topologies in fully informed and best-of-neighborhood particle swarms",*IEEE Transactions on Systems Man and Cybernetics Part C Applications and Reviews*, **vol. 36**, p. 515, 2006.

[3] X.-S. Yang, *Engineering optimization: an introduction with metaheuristic applications*: John Wiley & Sons, 2010.

[4] H. Pohlheim, "Examples of objective functions," *Retrieved*, vol. 4, p. 2012, 2007.

[5] J. Bansal, P. Singh, M. Saraswat, A. Verma, S. S. Jadon, and A. Abraham, "Inertia weight strategies in particle swarm optimization," in *Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on*, 2011, pp. 633-640.