2022 13th International Conference on Information and Communication Technology Convergence (ICTC) | 978-1-6654-9939-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICTC55196.2022.9952759

An Improved Particle Swarm Optimization Algorithm Based on S-shaped Activation Function for Fast Convergence

Muhammad Haris
Department of Electrical and Electronic Engineering
Hanyang University
Ansan, South Korea
haris737@hanyang.ac.kr

Dost Muhammad Saqib Bhatti
Department of Electrical and Electronic Engineering
Hanyang University
Ansan, South Korea
saqib@hanyang.ac.kr

Haewoon Nam
Department of Electrical and Electronic Engineering
Hanyang University
Ansan, South Korea
hnam@hanyang.ac.kr

***Abstract*—In this paper an Improve particle swarm optimization algorithm(IPSO) is proposed in which an S-shaped activation function, which is inspired by the neural networks is used to update the acceleration factors, which play a significant role in fast convergence of particles within a given search space. So, in order to keep parity between the exploration and exploitation this S-shaped activation function take into account both the distance of the particle to its Pbest(Personal best position) and from a particle to its Gbest(Global best position), That's how its enhance the convergence rate. The proposed Improved PSO algorithm with S- shaped activation function is tested on some famous complex benchmark functions and its is also compared with some well-known PSO variants.**

I. Introduction

Particle swarm optimization(PSO) is one such technique that has received significant attention because of its use in widely constrained and unconstrained function optimization problems. In the past one decade, a number of heuristic algorithms were proposed for effectively solving optimization problems. The particle swarm optimization(PSO) Algorithm was originally designed by kennedy and eberher [1].PSO is a metaheuristic algorithm that imitates the behavior of biological swarms. PSO algorithm can be easily implemented in those applications which deal with optimization problems [2]. In PSO Algorithm the swarm is made up of some light particles with velocities [3], each of which shows an attainable solution in an area. The use of PSO algorithms in different areas, such as in the rapid and flexible deployment of UAVs. In [4], training neural networks, Solving complex nonlinear problems, security, localization in WSNs [5]. and in robots path planning and tracking. Beside that, it can also be used in a number of optimization problems, e.g system identification [6] and controller design [7].

In the PSO algorithm, each particle is considered a candidate desirable solution to a given optimization problem. The particles fly in the search space of the optimization problem, PSO can achieve a quick survey of the given search territory and can connect the particles to the global and local optimization solution. In the particle swarm optimization algorithm, the velocity (v) and position (x) of particles are updated based on equations (1) and (2). (c_1) and (c_2) are acceleration factors or coefficients,(c_1) is cognitive component while (c_2) is social component.(r_1) and (r_2) are random variables Beside that, In Improved PSO algorithm the inertial factor ω is used to check and balance between the global and local search.

$$\begin{aligned} V_i^k = \omega V_i^{k-1} + c_1 r_1 (P_{best,_i} - X_i^{k-1}) \\ + c_2 r_2 (G_{best,_i} - X_i^{k-1}), \end{aligned} \tag{1}$$

$$X_i^k = X_i^{k-1} + V_i^k, \tag{2}$$

$$w = (w_{start} - w_{end}) * \frac{MAXiter - iter}{MAXiter} + w_{end}, \tag{3}$$

In Equation (1) V_i^k refers to the updated velocity of particles,V_i^{k-1} is the previous velocity of particles, $P_{best,_i}$ best position find by each particle so far, however, $G_{best,_i}$ is not for the individual particles but its the best among all particles. Moreover in equation (2) X_i^k is the

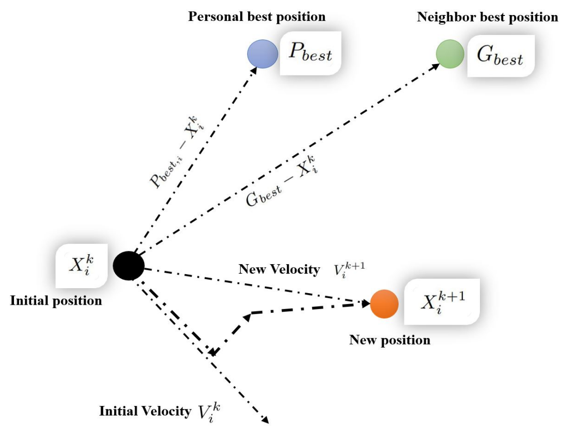

Fig. 1. Particle movement

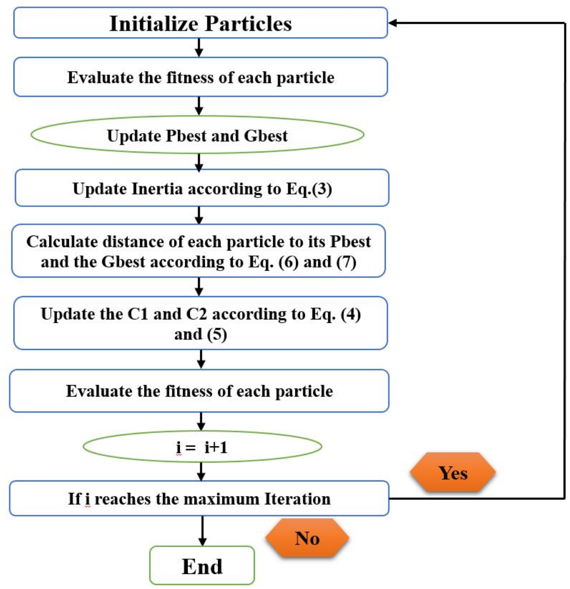

Fig. 2. The Flowchart of Improved PSO Algorithm

updated position of particles.

ω in equation (3) is the inertial weight factor. ω_{start} correspond to the inertial factor before the start of iteration while ω_{end} correspond to the inertial factor after a maximum number of iterations while Max_{iter} is the maximum number of iteration.

If we look at figure 1, there we have a particle (i) in time duration (t) which is located in position Xi(t), and its moves toward a vector Vi(t) but it is not allowed to move because it is a member of the swarm. Particles learn from each other, In addition to position and velocity, every particle has a memory of its best experience denoted by (Pi(t)). Besides the personal best, particles have common best experiences among a swarm known as global best denoted by (G(t)). So, we have a personal best position for every particle and we have a global best which is the best experience of all particles in a swarm. The numerical model of PSO is very simple, by defining the concept the velocity and position are updated after every single iteration and the newly updated position is (Xi(t+1)). The particle moves toward some new position using all the vectors like its personal best and global best. The new position is updated according to the previous velocity of Pbest and Gbest. it uses the previous experience of the whole swarm. The flowchart of the Improved sigmoid-based PSO algorithm is shown in figure 2.

The PSO algorithm was proposed back in 1995, Since then there are a number of variants being proposed to improve its performance.e.g PSO-LDIW[8]. In the PSO algorithm, the Inertial weight is used to balance the local and global search. A large inertial weight means better global exploration, However, a small inertial weight will encourage particles for exploitation[9]. Besides that, the problem with traditional PSO algorithms is local trapping and premature convergence. The PSO-LDIW algorithm has satisfactory performance in some of the applications. However, in the later part of the algorithm when the inertia weight decreases the searchability of the swarm is affected and it can not explore more new areas. Like inertia weight acceleration coefficient also compelled researchers to enhance the search capability of algorithm[10].In PSO-TVAC algorithm the cognitive acceleration coefficient (c_1) is linearly decreasing while the social acceleration coefficient (c_2) is linearly increasing. (c_1) and (c_2) pulls the particles to their respective personal and global position.

II. Proposed Algorithm

In this paper, an improved particle swarm optimization algorithm is proposed in which an S-shaped activation function is used to update the acceleration factor which helps the algorithm in fast convergence. The fast convergence happens because this activation function which is also called the sigmoid function, it takes into account both the particle distance from it personal best position and its distance from its global best position.In addition, based on these two distances it assign value to (c_1) and (c_2). Besides that, the sigmoid function that is used in this algorithm to update the acceleration factor is monotonic and bounded. It means that the function is monotonically increasing and evenly bound. Therefore, in this proposed algorithm if a particle is far away so it will give a larger (c_1) value to that particle so it will come quickly toward its personal best and global best position, and if a particle is near to global minima so it will assign the small value of (c_1) and large value of (c_2), due to which particle will explore search area near to global minima. That's how this algorithm increases the convergence rate, also this proposed algorithm keeps a balance between exploration and exploitation because these acceleration coefficients attract the particles toward their personal and global best positions. (c_1) and (c_2) are updated based on equation (4) and (5). The (H) in equation (8) measures the between particle and its personal best and the particle to its global best and then this (H) is input to the function. s, p, and a are the steepness of the curve, the peak value of the curve and the abscissa of the origin point respectively, In addition to that c is a positive constant value. It is very important to select the best values for these four parameters(s,p,a and c). As s is a parameter that defines the steepness of the curve, the value of a = 0.000040 is fixed and according to the behaviour of S-shaped activation function and from experimental behavior, p,a and c are set as 0.5, 0 and 1.3 respectively.

$$c_1(i) = F(Personalbest(k)), \tag{4}$$

$$c_2(i) = F(Globalbest(k)), \tag{5}$$

$$Personalbest(i) = P_{best,i-X_i^{k-1}}, \tag{6}$$

$$Globalbest(i) = G_{best,i-X_i^{k-1}}, \tag{7}$$

$$F(H) = \frac{p}{1 + exp^{-s\times(H-a)}} + c, \tag{8}$$

III. Experimental results and Discussion

In this paper, the current Sigmoid-based PSO algorithm is compared with some other popular variants of PSO algorithm, which include the basic traditional PSO[11], MDPSO(multi-model delayed particle swarm optimization)[12], and FOPSO(particle swarm optimization with fractional-order velocity)[13]are selected for the performance evaluation via five widely used benchmark functions. These different types of test functions are been simulated in Matlab. In figure 3 we can see that initially the particles are randomly distributed and

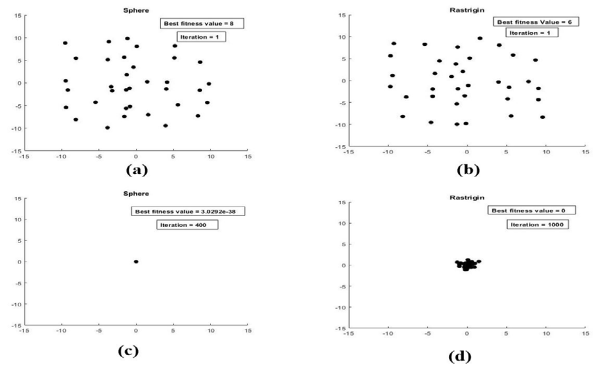

Fig. 3. Initial distribution of Particles

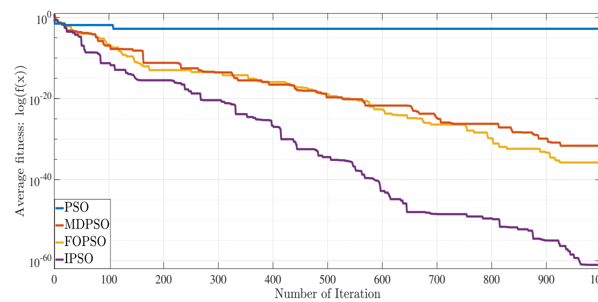

Fig. 4. Average Optimization performance of sphere function

in the PSO algorithm each particle is considered as a solution to a given problem. Therefore, these particles will search the area and after each iteration will move toward the optimal solution. The most important thing is to find the global optimum solution, for that simulation experiments are tested on Sphere, rosenbrock and Shwefel 2.21 and Griewank function In which Sphere and Shwefel 2.21 are uni-model benchmark functions which are most of time use to check the speed and accuracy of the optimization algorithm. However, griewank is a multi-model benchmark function and is used to check the ability of optimization algorithms to jump out of local trapping. The famous rosenbrock can be a multi-model or uni-model which depends on the cases of different dimensions. The initial and final distribution of the particles are shown in figure 3. The PSO algorithm is used to find the performance optimization curve of these functions. In Figure 4,5,6 and 7 the optimization curve of 4 different functions are shown when the PSO optimization method is used for the optimization process. For testing the optimization function that are used are as:

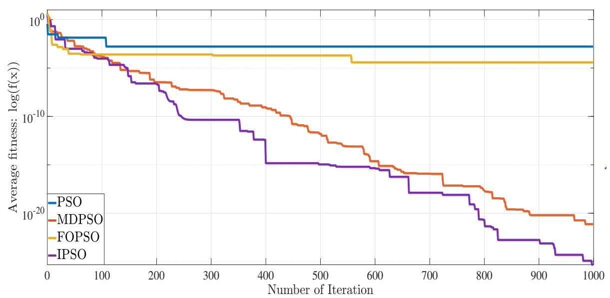

Fig. 5. Average Optimization performance of Schwefel 2.21 function

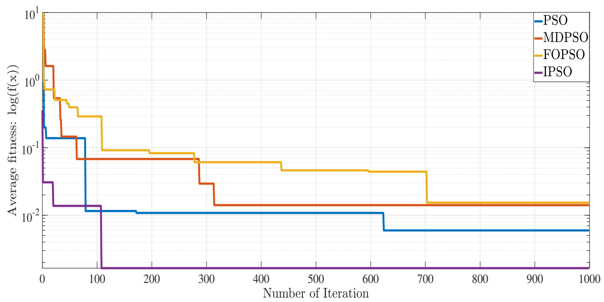

Fig. 6. Average Optimization performance of Rosenbrock function

$$f_1(x) = \sum_{i=1}^{n}(x_i^2) \tag{9}$$

$$f_2(x) = \sum_{i=1}^{n=1} \epsilon[100(x_{i-1} - x)^2 + (x_y - 1)^2] \tag{10}$$

$$f_3(x) = max_i(x_i, 1<=i<=D) \tag{11}$$

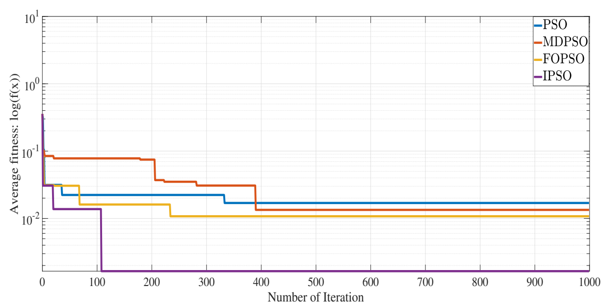

Fig. 7. Average Optimization performance of Griewank function

TABLE 1. Statistical results of related PSO Algorithm

Functions		*PSO*	*MDPSO*	*FOPSO*	*IPSO*
f1(x)	AOV	8.18e−30	3.33e−25	3.92e−38	3.33e−64
f2(x)	AOV	8.18e−25	3.33e−9	3.92e−20	3.33e−29
f3(x)	AOV	5.43e + 3	2.96	5.29	3.95e−2
f4(x)	AOV	7.74e−1	6.39e−1	3.09e−1	2.27e−24

TABLE 2. Configuration of Benchmark function

Functions	***Function name***	***Search space***	***Threshold***
f1(x)	Sphere	[−100, 100]	0.1
f2(x)	Schwefel2.21	[−100, 100]	0.1
f3(x)	Rosenbrock	[−30, 30]	10
f4(x)	Griewank	[−600, 600]	0.1

$$f_4(x) = -1/4000\sum_{i-1}^{n} x_i^2 - \Pi_{i=1}^{n} cos(x_i/\sqrt{i}) + 1 \tag{12}$$

The mentioned results can be confirmed by the statistical results in Table 1. Where the information about tested benchmark functions can be seen. It can be found that the proposed S-shaped activation based Improved particle swarm optimization(IPSO) has smaller average optimal value(AOV) than other functions.From that we can confirm that the proposed algorithm has fast convergence as compared to other variants of PSO.Moreover, the configuration of benchmark functions are shown in table 2.

IV. Conclusion

In this paper an improved PSO algorithm is proposed in which the acceleration coefficients are updated based on S- shaped activation function(Sigmoid). which measure the distance of particles in a swarm and based on that update the acceleration coefficients which help the particles to converge faster to optimal location.This also provide additional power to particles in case of local trapping. Thus this new IPSO algorithm is is proposed in term of new adaptive scheme. Moreover, the performance of the new Improved PSO is analyzed by some experiments on different benchmark functions and from the results we can confirm the superiority of the proposed algorithm.

Acknowledgement

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (2022R1A2C1011862) and Brain Pool (2021H1D3A2A0203932611).

References

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948 vol.4.

[2] A. Ratnaweera, S. Halgamuge, and H. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.

[3] M.-P. Song and G.-C. Gu, "Research on particle swarm optimization: a review," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, vol. 4, 2004, pp. 2236–2241 vol.4.

[4] Z. Yuheng, Z. Liyan, and L. Chunpeng, "3-d deployment optimization of uavs based on particle swarm algorithm," in *2019 IEEE 19th International Conference on Communication Technology (ICCT)*, 2019, pp. 954–957.

[5] Y. Yang, B. Li, and B. Ye, "Wireless sensor network localization based on pso algorithm in nlos environment," in *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, vol. 01, 2016, pp. 292–295.

[6] J. Liang, A. Qin, P. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.

[7] B. Song, Y. Xiao, and L. Xu, "Design of fuzzy pi controller for brushless dc motor based on pso–gsa algorithm," *Systems Science & Control Engineering*, vol. 8, no. 1, pp. 67–77, 2020.

[8] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, 1998, pp. 69–73.

[9] J. Liu, Y. Mei, and X. Li, "An analysis of the inertia weight parameter for binary particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 666–681, 2016.

[10] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, and R. G. Harley, "Particle swarm optimization: Basic concepts, variants and applications in power systems," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 171–195, 2008.

[11] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 1995, pp. 39–43.

[12] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.

[13] E. Pires, J. Tenreiro Machado, P. Moura Oliveira, J. Cunha, and L. Mendes, "Particle swarm optimization with fractional-order velocity," *Nonlinear Dynamics*, vol. 61, pp. 295–301, 07 2010.