

A Novel and Fast Memory Perturbation Method to Increase Exploration in Particle Swarm Optimization Algorithm

Muchamad Taufiq Anwar¹, Edy Winarno¹, Iswatun Chasanah², Saptono Nugrohadi³

muchamadtaufiqanwar@edu.unisbank.ac.id, edywin@edu.unisbank.ac.id, uswatunnana@gmail.com, sapton2000@gmail.com

¹Universitas Stikubank, Semarang, Indonesia

²Satya Wacana Christian University, Salatiga, Indonesia

³Universitas PGRI Semarang, Semarang, Indonesia

Abstract Particle Swarm Optimization (PSO), one of the versatile nature-inspired optimization algorithm, continue to suffer from premature convergence despite the numerous amount of research trying to improve this algorithm. Many research had tried to address this issue but often use a complex algorithm which tax on computational time and complexity. This research introduced a novel perturbation method to mitigate premature convergence / to increase exploration while keeping the computational cost at a minimum. The particles' memories (i.e the position of personal and global best) are modified by a random multiplier which in turn will 'perturb' the particles' velocity. The implementation of this novel perturbation method in early iterations had resulted in 100% success rate in finding global optima in multimodal benchmark tests including the Rastrigin problem – whereas the original PSO failed in all benchmark tests – without adding a significant amount of computational complexity and time.

Keywords: PSO, premature convergence, memory perturbation, exploration

1. Introduction

Particle Swarm Optimization (PSO) is a versatile optimization algorithm used in several areas. The original PSO was introduced in 1995 by Eberhart and Kennedy, and had gained popularity in research to further improve the algorithm, customized it to particular problems, and combined it with other algorithms.

Some of the PSO enhancement strategies include the use of multi-sub-populations[1], particle restart[2], combining PSO with other search techniques[1], using different neighborhood topologies[3], and by having a better initial population [4]. Despite the rigorous research in PSO, one of the main problems which still remains for PSO is premature convergence, especially in problem with many local optima.

Many research had addressed this issue by incorporating perturbation to increase diversification/exploration. This kind of modification often incorporate complex computation and thus increase its computational and time cost. This research proposed a novel and simple perturbation method to mitigate premature convergence / to increase diversification/exploration while keeping the computation overhead at its minimum.

2. Literature Review

2.1. The Original PSO

PSO algorithm is inspired by how birds or fish flock for food. Individual / candidate solutions in PSO are called 'particles'. Each particle will move randomly in the search space to find the best solution. In each iteration, particles will move closer to the best solution by learning from the experience of the particle itself (cognitive learning, c_1 , applied to personal best, x_i^*) and from other particle experiences (social learning, c_2 , applied to global best, x^g). The c_1 and c_2 are positive constants which usually set to = 2. The formula of particle movement or "velocity update" v_i at $t+1$ is shown in Equation 1.

$$v_i(t + 1) = v_i(t) + c_1 r_1 [x_i^*(t) - x_i(t)] + c_2 r_2 [x^g(t) - x_i(t)]$$

The search capability of PSO in finding optimum solution lies in the tuning of particle's exploration and intensification in their movements[2], [5]. Exploration is when the particles explore a wider range of possibility, while intensification is when the particles move 'slowly' to find a better solution near a good solution (i.e the particles' memory of global best / personal best).

2.2. PSO problem: premature convergence

Research had shown that PSO tends to converge quickly (prematurely)[6], [7]. This characteristic of PSO will consequently cause a failure in finding an optimal solution, especially in a multimodal problem – a problem with multiple "good solutions" and one optimum solution. In a multimodal problem, PSO will tend to prematurely converge to / "trapped" in the "good solution" / local optima. This problem with the original PSO is called premature convergence[3], [4], [7]–[11]. Analysis showed that the premature convergence and the ability to find best solution / global optimum of PSO are both probabilistic[7], [12].

There had been numerous research which attempted to alleviate this issue such as[8], [9]. Some of the common strategies include the use of inertia weight and perturbation (explained later). These research often use a complex algorithm. The persistent premature convergence problem might mean that PSO needs exploration in general. It had been noted that the performance of the PSO mostly relies upon inertia weight and optimal parameter setting [10], [13], [14]. These parameters will affect the exploration and intensification ability of particles in PSO, thus research in solving premature convergence will focus on these parameter tuning.

2.3. Solving The Premature Convergence Problem: Exploration and Intensification Tuning

2.3.1. The Inertia Weight Strategy

The first attempt to tune the exploration and intensification is to incorporate inertia weight (IW)[15], which is not present in the original PSO. The inertia weight (α or w) refers to the amount of the contribution of the previous particle velocity to its current velocity. Inertia weight is intended to balance the exploitation and exploration when the particles are searching for optimal solution in a search space[13]. [15] used a randomized IW and in 1998, Shi and Eberhart used a linearly decreasing inertia weight (LDIW) (α) into the algorithm[16].

$$v_i(t + 1) = \alpha v_i(t) + c_1 r_1 [x_i^*(t) - x_i(t)] + c_2 r_2 [x^g(t) - x_i(t)]$$

Where the value of α gradually decreases from α_{max} to α_{min} throughout the iterations:

$$\alpha(t) = \alpha_{max} - (\alpha_{max} - \alpha_{min}) \frac{t}{T}$$

When $w = 0$, particle speed will only be affected by the personal best and global best position. This means that the particles will immediately change their position to the best position once the best position is known. A small value of w will increase intensification (local search). In contrast, when w is high, the current velocity of the particle will be affected by the previous velocity (inertia). Such particles maintain the previous speed even though a better position is known. A high value of w will increase diversification (global search). To successfully find the optimum values (maximum or minimum), high diversification is needed at the beginning of the iteration, while intensification is needed later in the iteration[17]. This can be achieved simply by using a linearly decreasing inertia weight. However, research had pointed out that LDIW strategy will cause PSO to prematurely converge to local optimum in multimodal problems[18], [19]. Other research also used an adaptive selection of inertia weight in

which the value of inertia weight will be set according to certain criteria within the process of finding an optimum solution, such as in [20]. However, research also showed that LDIW is still a competitive strategy compared to other IW strategies if it's parameters are set properly[18].

Table 1. Research on inertia weight strategies in PSO

Reference	Inertia-weight strategy	Note
Y. Shi and R. Eberhart, 1998 [16]	Linearly decreasing	LDIW runs from 1.4 to 0
Bansal et al., 2011[19]	Linearly decreasing	Best LDIW runs from 0.9 to 0.4
Arasomwan, 2013[18]	Linearly decreasing	LDIW is still competitive against other variants
Farooq et al., 2017[4]	Linearly decreasing	LDIW in [19] applied twice
Nobile et al., 2018[14]	Self-tuning	Fuzzy logic
Agrawal, 2018[13]	Adaptive	Cumulative binomial probability
A. Agrawal and S. Tripathi, 2019 [15]	Adaptive	Binomial probability distribution

2.3.2. The Perturbation Strategy

Another attempt to alleviate premature convergence is by the use of ‘perturbation’. A perturbation is a way to ‘randomize’ the particles’ movement so that they can explore a more diverse candidate solution in the attempt to find the global optimum. The list of research in PSO perturbation strategy and the position of this research are shown in Table 2.

Table 2. Research on perturbation strategies in PSO

Reference	Object of perturbation	Perturbation method	Additional strategy
Zhihao Yuan et al., 2005[21]	Global best	Random number	Inertia weight jump threshold
S. Das, A. Konar, and U. K. Chakraborty, 2005[22]	Particles’ position vectors	Differentially perturbed Velocity	-
A. M. Zavala, A. H. Aguirre, and E. V. Diharce, 2007[23]	Personal best	C-perturbation (Differential Evolution) and M-perturbation	-
A. H. Aguirre, A. M. Zavala, E. V. Diharce, and S. B. Rionda, 2007[24]	Personal best	Same as [23]	Constraint handling
E. Yang, A. Erdogan, T. Arslan, and N. Barton, 2007[25]	Global best and personal best	Random number	-
Maeda, 2007 [26]	Inertia weight	Simultaneous perturbation based on particles’ gradient	-
Xinchao, 2010 [11]	Global best	Possibility theory	-
R. Kundu, S. Das, R. Mukherjee, and S. Debchoudhury, 2014[10]	Global best	Difference mean	Aging guideline, acceleration coefficients
L. Mengxia, L. Ruiquan, and D. Yong, 2016[27]	Swarm membership	Elitism	Anderson chaotic mapping
This research	Global best and personal best	Random number	Linear decreasing inertia weight

3. Method

In the standard PSO method, the "perturbation" is only applied to the difference between personal best and global best with the current particle position. The disadvantage of this method is that it limits the exploration capability of particles which can actually be greater. (Yang et. Al., 2007) proposed a new way to maintain diversification by applying direct ‘perturbation’ to the (current) personal best and global best in each iteration[25]. It uses two random number, r_3 and r_4 , whinge range from -2 to 2, $r \in [-2, 2]$. The formula for particles’ velocity in [25] is shown in Equation 2.

$$v_i(t+1) = \alpha v_i(t) + \frac{[(1+r_3)x_{pi}(t) - x_i(t)]}{3} + \frac{[(1+r_4)x_g(t) - x_i(t)]}{3}$$

Inspired by[25], we propose a perturbation $k*r$ which are applied to personal best and global best as shown in Equation 3. This modification will cause the particle to have a probabilistic movement within the enclosed area shown in Figure 2. With r is a random number between 0 to 1, $r \in [0, 1]$. When k is set to 2, the $(k*r)$ will produce perturbation that ranges from 0 to 2 and is applied as a multiplier for the personal best and global best. The perturbation is applied to both personal and global best as an attempt to further increase the exploration capability of the particles.

$$v_i(t+1) = \alpha v_i(t) + c_1 r_1 [k_1 r_3 x_i^*(t) - x_i(t)] + c_2 r_2 [k_2 r_4 x^g(t) - x_i(t)]$$

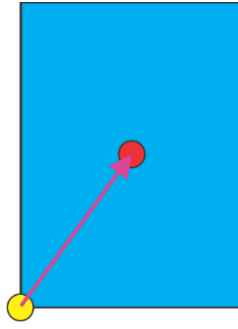


Figure 1. Comparison of particle movement in the original PSO (fuchsia arrow) vs the area of possible movement caused by perturbation (blue rectangle) where the yellow point is the previous best and the red point is the new best

We also experiment with a combination of IW proposed by Y. Shi and R.C. Eberhart [16] and use several range values of linear-decreasing iw. Here, we experiment with a w range that is somewhat wider (1-0) to see if the performance will get better. Experiments were done using Java programming language run in Netbeans. For all experiments, the parameter is set as follows: population= 50; max iteration= 100; $x_{max} = 100$; $x_{min} = -100$, $v_{max} = 100$, $v_{min} = -100$, $C1 = 2$, $C2 = 2$, $k_1 = 2$, $k_2 = 2$, and $r[0,1]$. IW is set with $\alpha_{max} = 1$ and $\alpha_{min} = 0$. Each experiment setting/type is repeated 1000 times. Then the success rate, min value and SD of min value for each test cases are calculated and compared.

4. Result

Based on 50 tests of each of the five function tests, the results of the combination of the Linearly-Decreasing Inertia-Weight (LDIW) and Memory Perturbation (MP) methods produced a 100% success rate in finding the minimum global for all test functions. This perturbation is only needed several times at the beginning of the iteration. Based on testing, the initial 10 iterations have given satisfactory results. The use of perturbation that is too long will actually weaken the power of LDIW's intensification. With 'nip' is the amount of perturbation we want at the beginning of the iteration (number of initial perturbation).

Table 3. Performance comparison between the original PSO, LDIW PSO, and LDIW-MP PSO

	Sphere	Schwefel 2.22	Rosenbrock	Rastrigin	Ackley
Standard PSO					
Mean	2,148973289	0,001496959	146,7146708	25,67040415	2,148973289
Std Dev	2,941967592	0,003330213	239,284953	17,058624	2,941967592
Success rate	0%	0%	0%	0%	0%
LDIW PSO					
Mean	0	0	0	0,119299392	0
Std Dev	0	0	0	0,323064281	0
Success rate	100%	100%	100%	88%	100%
LDIW MP-PSO					
Mean	0	0	0	0	0
Std Dev	0	0	0	0	0
Success rate	100%	100%	100%	100%	100%

The use of this LDIW-MP PSO method yielded satisfactory results with a 100% success rate in finding the minimum global in Sphere function, Schwefel Problem 2.22, Rosenbrock, and Ackley. But

for Rastrigin's function, the success rate is still around 88%. In the 12% of the experiments carried out, particles still trapped in local minima near the minimum global. This is understandable because the Rastrigin function does have many "traps" (local minima). This failure in the Rastrigin function is caused by the characteristics of the (standard) PSO where which the particles are "trapped" in a solution which appears to be the global minimum but is actually a local optimum.

To increase success in the Rastrigin function, an additional method is needed to maintain diversification so that particles can further explore the search space before determining the area for intensive searches. By maintaining diversification for some time, it is expected that particles can explore a more diverse search space and hopefully can visit an area near the minimum global – the intensification will proceed the digging into the minimum value.

The result showed that the best use of this perturbation method is to be applied only several times (10 times) in the initial iterations. Prolonged use of this perturbation will only result in particles moving randomly and failed to find global optimum, which is not what we want.

5. Conclusion

Despite a large amount of research, PSO still suffers from premature convergence where the particles are trapped in a sub-optimal solution in a multimodal problem. This research aims to explore a new strategy to increase the particles exploration capacity of particles in PSO in order to increase its capability to find the global optimum. Two perturbation factors are applied to global best and personal best to increase the exploration of search space in the hope of that the particle will stop at a point near the global optimum and continue to dig into the minimum value. The experimental result showed that this applying this perturbation method in early iteration, and combined it with LDIW can achieve 100% success rate in finding global optimum in five benchmark problems, i.e the Sphere function, Schwefel Problem 2.22, Rosenbrock, Rastrigin, and Ackley. Compared to other research, this research proposes a simple perturbation method which did not add a significant amount of computation complexity and time. The limitation of this study is that this method was only benchmarked against 2D problems. Further researches are open to experimenting on problems with a higher dimension.

References

- [1] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft Comput.*, vol. 22, no. 2, pp. 387–408, Jan. 2018.
- [2] D. Tamayo-Vera, S. Chen, A. Bolufé-Röhler, J. Montgomery, and T. Hendtlass, "Improved Exploration and Exploitation in Particle Swarm Optimization," 2018, pp. 421–433.
- [3] K.-L. Du and M. N. S. Swamy, "Particle Swarm Optimization," in *Search and Optimization by Metaheuristics*, Cham: Springer International Publishing, 2016, pp. 153–173.
- [4] M. U. Farooq, A. Ahmad, and A. Hameed, "Opposition-based initialization and a modified pattern for Inertia Weight (IW) in PSO," in *2017 IEEE International Conference on Innovations in Intelligent Systems and Applications (INISTA)*, 2017, pp. 96–101.
- [5] M. Oliveira, D. Pinheiro, M. Macedo, C. Bastos-Filho, and R. Menezes, "Better exploration-exploitation pace, better swarm: Examining the social interactions," in *2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, 2017, pp. 1–6.
- [6] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, pp. 1945–1950.
- [7] G. Xu, Z. H. Wu, and M. Z. Jiang, "Premature convergence of standard particle swarm optimisation algorithm based on Markov chain analysis," *Int. J. Wirel. Mob. Comput.*, vol. 9, no. 4, p. 377, 2015.
- [8] S. U. Khan, O. U. Rehman, N. Khan, A. Khan, S. A. A. Shah, and S. Yang, "Improving the Diversity of PSO for an Engineering Inverse Problem using Adaptive Inertia Weight," *Teh. Vjesn. - Tech. Gaz.*, vol. 25, no. 6, pp. 1631–1637, Dec. 2018.
- [9] F. A. P. Paiva, J. A. F. Costa, and C. R. M. Silva, "A serendipity-based pso approach to delay premature convergence using scout particle," *Int. J. Innov. Comput. Inf. Control*, vol. 12, no. 4, pp. 1349–4198, 2016.
- [10] R. Kundu, S. Das, R. Mukherjee, and S. Debchoudhury, "An improved particle swarm

- optimizer with difference mean based perturbation,” *Neurocomputing*, vol. 129, pp. 315–333, Apr. 2014.
- [11] Z. Xinchao, “A perturbed particle swarm algorithm for numerical optimization,” *Appl. Soft Comput.*, vol. 10, no. 1, pp. 119–124, Jan. 2010.
- [12] G. Xu and G. Yu, “Reprint of: On convergence analysis of particle swarm optimization algorithm,” *J. Comput. Appl. Math.*, vol. 340, pp. 709–717, Oct. 2018.
- [13] A. Agrawal and S. Tripathi, “Particle swarm optimization with adaptive inertia weight based on cumulative binomial probability,” *Evol. Intell.*, pp. 1–9, Nov. 2018.
- [14] M. S. Nobile, P. Cazzaniga, D. Besozzi, R. Colombo, G. Mauri, and G. Pasi, “Fuzzy Self-Tuning PSO: A settings-free algorithm for global optimization,” *Swarm Evol. Comput.*, vol. 39, pp. 70–85, Apr. 2018.
- [15] A. Agrawal and S. Tripathi, “Particle Swarm Optimization with Probabilistic Inertia Weight,” Springer, Singapore, 2019, pp. 239–248.
- [16] 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)*, pp. 69–73.
- [17] J. C. Bansal, P. K. Singh, M. Saraswat, A. Verma, S. S. Jadon, and A. Abraham, “Inertia Weight strategies in Particle Swarm Optimization,” in *2011 Third World Congress on Nature and Biologically Inspired Computing*, 2011, pp. 633–640.
- [18] M. A. Arasomwan and A. O. Adewumi, “On the performance of linear decreasing inertia weight particle swarm optimization for global optimization.,” *ScientificWorldJournal.*, vol. 2013, p. 860289, Oct. 2013.
- [19] J. C. Bansal, P. K. Singh, M. Saraswat, A. Verma, S. S. Jadon, and A. Abraham, “Inertia Weight strategies in Particle Swarm Optimization,” in *2011 Third World Congress on Nature and Biologically Inspired Computing*, 2011, pp. 633–640.
- [20] H. D. Purnomo and H.-M. Wee, “Particle swarm optimisation with adaptive selection of inertia weight strategy,” *Int. J. Comput. Sci. Eng.*, vol. 13, no. 1, pp. 38–47, 2016.
- [21] Zhihao Yuan *et al.*, “A Perturbation Particle Swarm Optimization for the Synthesis of the Radiation Pattern of Antenna Array,” in *2005 Asia-Pacific Microwave Conference Proceedings*, vol. 3, pp. 1–4.
- [22] S. Das, A. Konar, and U. K. Chakraborty, “Improving particle swarm optimization with differentially perturbed velocity,” in *Proceedings of the 2005 conference on Genetic and evolutionary computation - GECCO '05*, 2005, p. 177.
- [23] A. M. Zavala, A. H. Aguirre, and E. V. Diharce, “Robust PSO-based constrained optimization by perturbing the particle’s memory,” in *Swarm intelligence, focus on ant and particle swarm optimization*, IntechOpen, 2007.
- [24] A. H. Aguirre, A. M. Zavala, E. V. Diharce, and S. B. Rionda, “COPSO: Constrained Optimization via PSO algorithm,” *Cent. Res. Math. (CIMAT). Tech. Rep. No. I-07-04/22-02-2007*, vol. 77, 2007.
- [25] E. Yang, A. Erdogan, T. Arslan, and N. Barton, “An Improved Particle Swarm Optimization Algorithm for Power-Efficient Wireless Sensor Networks,” in *2007 ECSIS Symposium on Bio-inspired, Learning, and Intelligent Systems for Security (BLISS 2007)*, 2007, pp. 76–82.
- [26] Y. Maeda and N. Matsushita, “Simultaneous Perturbation Particle Swarm Optimization Using FPGA,” in *2007 International Joint Conference on Neural Networks*, 2007, pp. 2695–2700.
- [27] L. Mengxia, L. Ruiquan, and D. Yong, “The Particle Swarm Optimization Algorithm with Adaptive Chaos Perturbation.,” *Int. J. Comput. Commun. Control*, vol. 11, no. 6, 2016.