A Comparative Study on Different Parameter Factors and Velocity Clamping for Particle Swarm Optimisation

S P Lim1,*, H Hoon2, and W H Song1

1Department of Computer Science, Faculty of Information and Communication Technology, Universiti Tunku Abdul Rahman, Jalan University, Bandar Barat, 31900 Kampar, Perak, Malaysia.
2Department of Finance, Faculty of Business and Finance, Universiti Tunku Abdul Rahman, Jalan University, Bandar Barat, 31900 Kampar, Perak, Malaysia.

Email: limsp@utar.edu.my, lawrencess87@yahoo.com

Abstract. Optimisation is an interesting topic because various optimum results can be obtained using limited resources. This is the reason that researchers are interested in further investigating the possibility of optimisation by testing numerous optimisation methods. As the current research trends are preferred to solve the problems using Artificial Intelligence (AI) methods, hence more nature-inspired optimisation methods are developed. Particles Swarm Optimisation (PSO) is one of the well-known optimisation methods in Swarm Intelligence (SI). It can be used to solve many problems. However, PSO is easily trapped in local optimum and encountered premature convergence problem, which indirectly affects its performance. Besides that, its performance with different parameter factors and velocity clamping is always not considered by researchers. Therefore, the objective of this research is to compare the performance of PSO with different parameter factors and velocity clamping using average fitness values. 10 different types of PSO models are applied and tested using 10 benchmark functions. Based on the experimental results, PSO10 with constriction factor and velocity clamping is proved to be the best PSO model because it can obtain 10 out of 10 minimum average fitness values. The outcomes can assist the researchers while applying the method to their research.

1. Introduction

Optimisation is an active and interesting research area [1] to be explored because it can produce optimum results using minimum resources. Researchers try to solve the same optimisation problems using different methods because they want to investigate the performance of the method before incorporating it into their research [2]. This is to ensure the method can achieve good performance [2]. Besides that, the issue and limitation of the method can also be explored when testing with different case studies. Otherwise, poor performance will be obtained when applying on the complicated case studies. As mentioned by [3], traditional mathematical optimisation methods are less effective in solving challenging problems. Those methods easily fall into the suboptimal solution and a longer time is needed to obtain the optimum result. As stated by [1, 4], when the real-world problems and stochastic case studies are getting complicated, alternative and good optimisation methods are needed to produce an excellent result. Hence, the result will be influenced by the type of problems or the methods applied [5]. Recently, most of the new methods are developed and inspired based on nature. Yet, Particle Swarm Optimisation (PSO) is still one of the famous nature-inspired methods for AI [2] and SI [6]. It was proposed by Kennedy and Eberhart in 1995 [2, 7, 8]. It is a population-based method, which is similar to Evolutionary Computation (EC) methods and developed based on the social behaviour of animals [6]. It is also frequently used in SI [9, 10] to solve various types of optimisation problems, such as engineering [11], surface reconstruction [12], and risk management [13]. As stated by [1, 6, 10, 14, 15], PSO is easy to implement, performs well and contains simple operation with a good convergence rate using the iterative process to update the population. So, it is suitable to solve real-world optimisation problems [4].

Although PSO is better compared to other SI methods, it is easily trapped in local optimum and encountered premature convergence problem [1, 4, 6, 16, 17]. Hence, as shown in [1, 16, 17, 18, 19], most of the previous works try to raise its performance [15] by enhancing with different methods and variables. Also, comparison of the performance of PSO has been done extensively in [1, 4, 6, 8, 15, 18, 20, 21] when the new concept has been proposed on PSO. However, some of the previous works only compared the results with the original version of PSO or without considering other versions of PSO, such as PSO with inertia weight (constant, linearly decreasing, random) and constriction factor. As shown in [4, 8], different inertia weight (w) and constriction factor (k) values can affect the performance of PSO. In addition, velocity clamping (vc) in PSO has also been neglected and underestimated by the previous works. It plays the main role in tuning the maximum velocity of the particles, which can avoid the particles from rapid acceleration. As stated by [22], different settings of maximum velocity will be able to affect the performance
of PSO. Besides that, the performance of PSO model without velocity clamping also shown some differences in the result produced. As mentioned by [23], the techniques in the operations are one of the issues that affect the performance of a method besides parameter setting. Hence, it proves that parameter factors (inertia weight and constriction factor) and velocity clamping can affect the performance of PSO. So, different methods should be applied and tested using the same problems and parameter setting [2] to obtain a better result. Therefore, the objective of this research is to compare the performance of PSO with different parameter factors and velocity clamping using benchmark functions.

2. Flow of Experiment

All the PSO models in this research will adopt the flow as discussed in [2]. It starts by generating a swarm of particles and follows by updating the position and velocity of particles. The value encoding scheme is applied to the particle. The length of the particles is determined by the dimension size of benchmark function, \( d \) whereas the parameter values, \( x_i \) in each particle are derived based on the range of benchmark function. The velocity of each particle is initialised with zero at the beginning of the population [24]. After the population has been generated, the values will be optimised using the PSO models. Table 1 shows the PSO models that are applied in this research. The content of the models can be referred to [7, 10, 11, 20, 25]. The major differences across various types of PSO models are with and without parameters factors (inertia weight or constriction factor), and also with and without velocity clamping. Hence, there are 10 different types of PSO model. Similar to [2, 5, 23], standard parameters are applied because the main focus of this research is to compare the performance of PSO models with different parameter factors and velocity clamping in generating the minimum average fitness values. Table 2 shows the parameters set which are referred from the previous works in [2, 4, 5, 8, 15, 16, 17, 20, 21, 23, 26]. For termination criteria, maximum generation is used [2, 16, 23] to terminate the model at the same iteration. The benchmark functions in [5, 27, 28] will be served as the fitness function for each PSO model. The actual optimum value for each benchmark function is 0 and there is no standard dataset for benchmark function as the values are randomly derived based on the range given [2, 5, 23]. Kindly refer to [2, 5, 23, 27, 28] for the details of the benchmark function. All the benchmark functions are tested 10 times [2, 5, 23, 26] in generating the average fitness values. The reason is to show that different fitness values will be generated using the same model. Therefore, the model which is capable to generate minimum average fitness values and approximate towards the actual optimum value is recognised as the best model with good performance. All the experiments are performed using Dev C++.

### Table 1. Types of PSO model

| PSO Model | Combination |
|-----------|-------------|
| 01 | Original PSO |
| 02 | Original PSO with vc |
| 03 | PSO with constant w |
| 04 | PSO with constant w and vc |
| 05 | PSO with linearly decreasing w |
| 06 | PSO with linearly decreasing w and vc |
| 07 | PSO with random w |
| 08 | PSO with random w and vc |
| 09 | PSO with k |
| 10 | PSO with k and vc |

### Table 2. Parameter Setting

| No | Parameter | Value | References |
|----|-----------|-------|------------|
| 1 | Number of Generation | All | 2000 | [2, 5] |
| 2 | Population Size | All | 40 | [2, 5, 17] |
| 3 | Number of Dimensions | All | 30 | [2, 5, 17] |
| 4 | PSO Random Number (rand1, rand2) | All | [0.1] | [2, 16, 20] |
| 5 | PSO Maximum Velocity | PSO (02, 04, 06, 08, 10) | half of the range of the dataset | [2] |
| 6 | PSO Maximum Position | All | the range of the dataset | [2] |
| 7 | PSO Acceleration Constant (c1 and c2) | PSO01 – PSO08 | 2 | [2, 17, 26] |
| 8 | Inertia Weight (\( w \)) | PSO09 – PSO10 | 0.95 | [2, 4, 21] |
| 9 | Constriction Factor (\( k \)) | PSO03 and PSO04 | 0.7 | [2] |
| 10 | Number of Testing | All | 10 | [2, 5, 23, 26] |

### 3. Analysis and Discussion

All soft computing methods do not contain mathematical evidence in testing the convergence and obtaining an accurate result [2, 5]. Besides that, not all case studies can be formulated in a mathematically way [29] and solved using PSO because different results will be generated along with various computation times. So, the statements are also valid on PSO. Similar to the previous works [2, 4, 5, 23], the experimental results are used to compare the performance of miscellaneous types of PSO models in generating the minimum average fitness values. Table 3 presents the average fitness values generated by each PSO model. As shown in Table 3, PSO10 contains the best performance compared to the others because it can achieve 10/10 minimum average fitness values for different benchmark functions. As this research is focused on the optimisation towards the actual optimum for benchmark function, so the accuracy is given priority instead of speed.

As shown in Table 3, the results can be further improved when inertia weight (PSO03 – PSO08) and constriction factor (PSO09 and PSO10) are added into PSO01 and PSO02 models, which are the original version of PSO. Hence, with the addition of inertia weight or constriction factor, it can raise the performance of PSO. Inertia weight plays the main key role in the performance of PSO [20] whereas constriction factor PSO can provide better quality results compared to the original PSO [11]. Besides that, when the technique of inertia weight is changing, it can further optimise the results, as shown in PSO03 to PSO08. When inertia weight is in constant value (PSO03 and PSO04), it is
better than the original version PSO (PSO01 and PSO02) because it can manage the velocity of particles. As informed by [18], when the inertia weight is constant, the higher solution can escape from the minimal point whereas the lower solution tends to converge. However, the particle is unable to explore the search space in a flexible way due to the variation of velocity is limited. So, when inertia weight is linearly decreasing (PSO05 and PSO06) from higher to smaller value, the variation of velocity is increasing, which indirectly improves the result compared to PSO03 and PSO04. It shows that when inertia weight is high, PSO will perform the global search and when inertia weight is low, it will perform the local search, which also improves the results. This is also supported by [4, 8, 17, 19, 30]. Besides that, when inertia weight is randomly generating, the velocity will be randomly tuned, which can improve the search space in the population. As shown in PSO07, the results produced are better compared to PSO03 and PSO05. Yet, PSO08 is better than PSO02, but not PSO04 and PSO06. It will be explained along with velocity clamping. As shown in Table 3, when the inertia weight has been removed and the construction factor has been added, PSO09 and PSO10 performed even better compared to PSO03 to PSO08. This is because the constriction factor can avoid the drastic change in the position of the particle by controlling the influence of velocity, personal and global best position. As mentioned by [11], the constriction factor can guarantee the convergence of the search procedure. Hence, it can generate better results compared to inertia weight.

Table 3. Experimental Results

| Function   | PSO01 | PSO02 | PSO03 | PSO04 | PSO05 | PSO06 | PSO07 | PSO08 | PSO09 | PSO10 |
|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Sphere     | 987261.100 | 249034.700 | 953980.200 | 142172.980 | 947460.100 | 127131.090 | 930970.400 | 158725.620 | 919259.900 | 105656.200 |
| Ackley     | 128.661 | 9.353 | 112.593 | 0.069 | 105.703 | 0.027 | 91.856 | 0.149 | 89.324 | 0.009 |
| Rastrigin   | 128.661 | 9.353 | 112.593 | 0.069 | 105.703 | 0.027 | 91.856 | 0.149 | 89.324 | 0.009 |
| Griewank    | 466.711 | 265.367 | 463.795 | 64.911 | 450.302 | 50.979 | 429.316 | 118.638 | 423.664 | 43.387 |
| Zakharov    | 2710.673 | 673.203 | 2548.841 | 0.374 | 2473.991 | 4.04E-08 | 2412.364 | 10.098 | 2319.978 | 1.89E-13 |
| Shekel      | 1411.365 | 157.990 | 1039.992 | 43.111 | 827.300 | 31.777 | 653.473 | 52.566 | 544.517 | 5.41E-09 |
| Schwefel22  | 7872.848 | 18.760 | 7777.196 | 58.871 | 7756.800 | 28.948 | 7731.907 | 28.448 | 7731.083 | 28.448 |

Besides that, velocity clamping can also affect the performance of PSO. As shown in Table 3, PSO01, PSO03, PSO05, PSO07, and PSO09 are the models without velocity clamping whereas the remaining models contain the velocity clamping. For the models with velocity clamping, the results are further optimised compared to the models without velocity clamping. As stated by [11], the quality of results for PSO will be affected by cognitive and social parameters, as well as velocity limits. As stated in [8], when maximum velocity is equal to the maximum range value of the position, it can improve the performance of PSO. For the models without velocity clamping, PSO09 is performed better compared to the others. Similar goes to PSO07, it is better than PSO05 and PSO05 is better than PSO03. As for the models with velocity clamping, PSO10 is performed better than PSO08 whereas PSO06 is better than PSO04. Yet, PSO08 is not performed well compared to PSO04 and PSO06. This is maybe the random inertia weight affects the velocity of particles by producing higher velocity compared to maximum velocity. Hence, it is unable to outperform compared to the inertia weight which is constant and linearly decreasing. As stated by [8], maximum velocity can be excluded when applying the constriction factor and the particles are allowed to fly outside the range of the maximum position. As shown in Table 3, PSO10 with constriction factor and velocity clamping can further optimise the values compared to the remaining models. So, velocity clamping can control the velocity and position of each particle, which indirectly influences the performance of PSO [2, 31].

4 Conclusion and Future Work

PSO10 with constriction factor and velocity clamping is proved to be the best combination of the PSO model. It manages to obtain the highest number of minimum average fitness values. For the constriction factor, it can guarantee the convergence of PSO [32] whereas velocity clamping can control the range of velocity and prevent the step size from getting larger [34]. It is noticed that different techniques in generating inertia weight can produce different results. Hence, it is recommended the researcher to test case studies with various types of PSO models in order to obtain a better result. For future work, it is suggested to look into the parameters inside PSO. As proposed by [2], the number of parameters can be reduced, such as only focusing on the global best position in updating the position of particles. By using the models as suggested in this research, various parameter values can be tested using more benchmark functions in order to generate excellent results. Innovation on the techniques or hybrid methods in generating inertia weight or constriction factor can be proposed. This is because as shown in Table 3, when the techniques used in generating the inertia weight are different, the performance of PSO will be affected. When the best PSO model has been obtained, it can also be compared with the other SI or AI methods, which is also supported by [21]. Finally, the PSO model can be used to solve the real-world case studies, such as in surface reconstruction [12] as supported by [14, 33] and also in risk management [34].

Acknowledgment

This work is financially supported by FRGS from Ministry of Education, Malaysia.

References

[1] J J Liang, A K Qin, P N Suganthan, S Baskar 2006 IEEE Trans. Evo. Comp. 10 pp 281
[2] S P Lim, H Haron 2013 ICOS 2013, 41
[3] G G Wang, Y Tan 2017 IEEE Trans. on Cyber. 1
[4] G Singh, A Singh 2017 Int. J. of Eng. Res. & Tech. S, 1, 1
[5] S P Lim, H Haron 2013 ACIDS 2013 LNAI 7802 pp 255
[6] A Uriarte, P Melin, F Valdez 2016 IEEE 8th ICIS pp128
[7] J Kennedy, R C Eberhart 1995 IEEE ICNN pp 1942
[8] R C Eberhart, Y Shi 2000 CEC99 pp 84
[9] G Yang, X Yuan 2007 ICNNMC pp 5672
[10] L Cao, L Xu, E D Goodman 2016 Comp. Intel. & Neuro 2016 1
[11] B B Raju, C P Raju, R S Rao 2018 Int. J. Elect. Eng. & Info. 10 pp 232
[12] A Gálvex, A Cobos, J, Puig-Pey, An Iglesias 2008 ICCS 2008, LNCS 5102 pp 116
[13] Q. Huang, 2013 ICCIS, 670 (2013)
[14] S P Lim, H Haron 2014 Art. Int. Rev. 42 pp 59
[15] X Yang, Q Jiao, X Liu 2019 IEEE 3rd ITNEC pp 2084
[16] S Biswasraha, K K Mandal, N Chakraborty 2013 Pro. Ener 7 pp 174
[17] M Alhussein, S I Haider 2015 3rd ICAIMS pp 61
[18] F Olivas, F Valdez, O Castillo 2013 WCNApBIC pp 36
[19] K E Parsopoulos, V P Plagianakos, G D Magoulas 2011 Adv. Con. Anal. & Glo. Opt. pp 445
[20] R C Eberhart, Y Shi 2001 CEC01 pp 94
[21] Z H Zhan, J Zhang, Y Li, H S H Chung 2009 IEEE Trans. Sys. Man Cyber. P B: Cybern. 39 pp 1362
[22] M Pluhacek, R Senkerik, A Viktorin, T Kadavy 2018 32nd ECMS 1
[23] S P Lim, H Hoon, C H Ong 2018 8th ICCSCE, 59
[24] A Engelbert 2012 IEEE WCCI 1
[25] Y Shi, R Eberhart 1998 IEEE ICEC 69
[26] A H Al-mter, S F Lu 2016 ICISAI 2016 pp 432
[27] S Surjanovic, D Bingham 2019 https://www.sfu.ca/~ssurjano/optimization.html. [Online]. Accessed on 1 Dec 2019
[28] D Eriksson 2019 https://codedocs.xyz/dme65/SOT/classsot_1_1Schewefel22.html [Online] Accessed on 1 Dec 2019.
[29] F H Grupe, S Jooste 2004 Info. Man. & Comp. Sec. 12 pp 288
[30] L Wang, J Zhang, X Li, G Sun 2015 27th CCDC pp 1200
[31] J Kennedy, R C Eberhart, Y Shi 2001 Swa. Intel.
[32] M Clerc 1999 CEC99, pp1951
[33] S P Lim, H Haron 2014 ICISP 2014 pp 602
[34] Z Yu, Q Mao, W Wang 2010 ICAICI 2010 3 pp 444