2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

A Cautionary Note on Poli's Stability Condition for Particle Swarm Optimization

1st Daniel H. von Eschwege
Department of Industrial Engineering
Stellenbosch University
Stellenbosch, South Africa
ORCID: 0000-0003-0935-9493

2nd Andries P. Engelbrecht
Department of Industrial Engineering and Computer Science Division
Stellenbosch University
Stellenbosch, South Africa
Center for Applied Mathematics and Bioinformatics
Gulf University for Science and Technology
Kuwait, Kuwait
ORCID: 0000-0002-0242-3539

***Abstract*—Particle swarm optimization (PSO) is a swarm intelligence algorithm that finds candidate solutions by iteratively updating the positions of particles in a swarm. PSO performance depends on the use of a suitable control parameter (CP) configuration, which governs the trade-off between exploration and exploitation in the swarm. Various methods of adapting or tuning CPs exist, but many result in exploding particle velocities and an unstable search process. Poli's stability condition ensures convergence in the mathematical limit, and is often used to inform CP configuration. However, this study shows that since it does not place any practical convergence constraints, it cannot be used to guarantee a stable search process. Velocity explosion occurs nonetheless and can lead to floating-point overflow and numerical instability. The investigation into various CP configurations across diverse functions and measurements of particle velocities provides empirical evidence of velocity explosion, and cautions against the assumption that enforcing Poli's criterion guarantees stability. The findings underline the need for comprehensive understanding of CP tuning and stability conditions in PSO, as well as the crucial role of empirical evidence in evaluating the real-world performance of swarm intelligence algorithms.**

***Index Terms*—Particle swarm optimization, Stability condition, Poli, Swarm intelligence**

I. Introduction

Particle swarm optimization (PSO) is a form of swarm intelligence algorithm, initially proposed by Kennedy and Eberhart [1], and draws inspiration from the congregation patterns of bird flocks. The PSO method finds potential solutions through continuous modification of the position and velocities of particles within a swarm. The momentum of each particle, the best position it has visited, and the best position visited by any particle in its neighbourhood are used to calculate these changes in position and velocity.

PSO is suitable for multiobjective optimization problems, and complicated problems with deceptive fitness landscapes where standard gradient-based algorithms are prone to become stuck in a local minimum. The effectiveness of PSO is reliant on the usage of an appropriate control parameter (CP) configuration, which balances exploration and exploitation within the swarm [2]–[5]. Furthermore, the CP configuration dictates whether the swarm approaches an equilibrium or diverges.

The No Free Lunch theorem [6] states that any two algorithms are equivalent when their performance is assessed as an average over all problems, which extends to CPs in that no CP configuration is optimal for all problems. Therefore, the CP configuration must be tuned for the problem at hand if optimal performance is desired [3], [7]. Furthermore, if the particle swarm diverges and leaves the search space permanently, no good solutions are found.

To prevent divergence, various stability conditions have been derived, all defining convergence regions from which stable CP values can be sampled. Examples of stability conditions are those by Van den Bergh [3], Kadirkamanathan [8], Gazi [9], and Poli [10]. Poli's criterion was also derived in an assumption free manner [11]–[15], and has empirically been demonstrated to be the most accurate amongst the various stability conditions [16], [17]. Poli's criterion therefore provides useful guidance with regards to setting CP configurations conducive towards convergence, defined in this context as an equilibrium state where particles do not move anymore, not that they necessarily end up at the same point in the landscape.

However, while Poli's criterion ensures convergence in the mathematical limit, it does not place any practical convergence constraints. The criterion does not specify that convergence will occur within K time-steps, or that particle step sizes will remain below some bound V_{max}. PSO can only ever be run within a finite computational budget and with a finite floating point precision. Therefore, PSO search often diverges in a practical sense when considering the computational budget constraints, even when CPs lie well within the convergence region defined by Poli's criterion. Nonetheless, an equilibrium may still be reached if the search continues in the limit.

This paper emprically shows velocity explosion in PSO, despite satisfaction of Poli's stability condition. The PSO search behaviour is investigated by stepping through CP configurations for various functions, and measuring particle velocities during the search. This study thereby cautions against the blind assumption that enforcing Poli's criterion will necessarily ensure a computationally stable search process under limited computational budgets, as doing so could well

978-0-7381-4408-5/23/$31.00 ©2023 IEEE

result in numerical instability and floating-point overflow. The contribution is therefore to gain a better understanding of the practical implications of PSO stability conditions.

The remainder of this paper is structured as follows: Section II provides background information on PSO and Poli's stability condition. Section III describes the experimental setup and methodology. Section IV presents the results of the experiments, and Section V concludes the paper.

II. Background

This section elucidates the PSO algorithm and its CPs, as well as Poli's stability condition.

A. Particle swarm optimization

PSO employs swarm search to address optimization problems. It modifies particle velocities and subsequently their positions on each time-step. This is based on the best position found by a particular particle, the best position discovered by the particle's neighbourhood [1]. The inertia weight PSO variant also takes into account the particle's previous velocity [18] The velocity update equation is expressed as

$$\begin{aligned} v_{ij}(t+1) = \omega v_{ij}(t) + c_1 r_{1ij}(t)\left[y_{ij}(t) - x_{ij}(t)\right] \\ + c_2 r_{2ij}(t)\left[\hat{y}_{ij}(t) - x_{ij}(t)\right] \end{aligned} \quad (1)$$

where at a certain time-step t, particle i in dimension j possesses a velocity $v_{ij}(t)$, a position $x_{ij}(t)$, a personal best position of $y_{ij}(t)$, and a neighbourhood best position of $\hat{y}_j(t)$.

The PSO CPs consist of the inertia weight ω, the cognitive acceleration coefficient c_1, and the social acceleration coefficient c_2. The CPs govern the trade-off between exploration and exploitation in the swarm, and have bearing on the convergence properties of the swarm [2]–[5]. Random elements r_{1j} and r_{2j} are sampled uniformly from the range (0,1), to prevent the swarm from cycling through the same positions repeatedly. Position updates are performed using

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \quad (2)$$

where $\mathbf{v}_i$ signifies the velocity vector and $\mathbf{x}_i$ signifies the position vector of particle i. The PSO pseudocode is given by Algorithm 1.

B. Control Parameter Configurations

PSO performance and behaviour is extremely sensitive to the CP configuration used. Even small changes in CPs can have a significant impact on the search process [4]. The optimal CP configuration also usually is problem-specific [19]. Furthermore, a CP configuration which is suitable at the current time step may also not be suitable at a later time step [20].

CP values can be fixed, often with the values $\omega = 0.729844$, $c_1 = 1.496180$, and $c_2 = 1.496180$ [19]. However, appropriate adjustment of CPs enables fine-grained control over the exploration-exploitation trade-off to improve PSO performance [4]. This trade-off is crucial torwards adequately exploring uncharted areas of the fitness landscape, while avoiding excessive computation on unpromising areas. In an attempt to dynamically control this trade-off based on the function landscape at hand, various approaches towards self-adaptive PSO (SAPSO) have been proposed [19]. SAPSO has the advantage that CPs do not need to be tuned anew for each problem, but rather are adjusted during the search process. Since SAPSO algorithms dynamically adapt CPs during the search process, it often occurs that the selected CPs result in velocity explosion and an unstable search processes [19], [21].

Algorithm 1 PSO Algorithm

```
Let f be the objective function of the optimization problem.
Uniformly initialize a swarm of n_s particles in n_d dimensions within the boundaries of the search space.
Initialize the personal best position y_i to x_i(0).
Initialize the neighbourhood best position ŷ_i to x_i(0).
Initialize v_i(0) to 0.
repeat
    for all particles i = 1, ..., n_s do
        if f(x_i) < f(y_i) then
            y_i = x_i
        end if
        for all particles î with neighbouring particle i do
            if f(y_i) < f(ŷ_î) then
                ŷ_î = y_i
            end if
        end for
    end for
    for all particles i = 1, ..., n_s do
        update v_i with the velocity update equation (1)
        update x_i with the position update equation (2)
    end for
until stopping condition reached
```

C. Stability Condition

To address the problem of unstable PSO search which results from velocity explosion, a number of stability conditions have been proposed, and are generally derived using certain assumptions. Given a stability condition, CP configurations which conform to the condition can then be sampled in the hope that this will result in a stable search process. A number of stability conditions are considered in this paper:

Van den Bergh [3] derived the following stability condition:

$$c_1 + c_2 < 2(1+w), \quad c_1 > 0, \quad c_2 > 0, \quad 0 < w < 1. \quad (3)$$

using the stagnation assumption,

$$\begin{aligned} \boldsymbol{y}_i(t) &= \boldsymbol{y}_i \ \forall \ t, \\ \hat{\boldsymbol{y}}_i(t) &= \hat{\boldsymbol{y}}_i \ \forall \ t, \end{aligned} \quad (4)$$

and the deterministic assumption,

$$\begin{aligned} \boldsymbol{\theta}_1 = \boldsymbol{\theta}_1(t) = c_1 \boldsymbol{r}_1(t) \ \forall \ t, \\ \boldsymbol{\theta}_2 = \boldsymbol{\theta}_2(t) = c_2 \boldsymbol{r}_2(t) \ \forall \ t, \end{aligned} \quad (5)$$

Derived by Kadirkamanathan under the stagnation assumption, the following provides a potential region for particle convergence [8], [16]:

$$\begin{cases} c_1 + c_2 < 2(1+\omega) & \omega \in (-1, 0] \\ c_1 + c_2 < \frac{2(1-\omega)^2}{1+\omega} & \omega \in (0, 1). \end{cases} \tag{6}$$

Also under the stagnation assumption, Gazi [9], [16] derived the stability condition:

$$\begin{cases} c_1 + c_2 < \frac{24(1+\omega)}{7} & \omega \in (-1, 0] \\ c_1 + c_2 < \frac{24(1-\omega)^2}{7(1+\omega)} & \omega \in (0, 1) \end{cases} \tag{7}$$

Finally, also under the stagnation assumption, Poli [10], [22] derived the stability condition:

$$c_1 + c_2 < \frac{24\left(1-\omega^2\right)}{7-5\omega} \text{ and } \omega \in [-1, 1] \tag{8}$$

which is diplayed in Figure 1.

Poli's criterion has also been derived in an assumption free manner [11] at a later stage. The criterion has also empirically been shown to be the most accurate [16], [17]. Poli's criterion guarantees that, in the limit, particle step sizes will converge to zero. Here, convergence is used to imply order-2 stability [10] where particles do not move anymore. Random variable sequences z_n are order-2 stable if

$$\lim_{n\to\infty} E\left[z_n\right] = \mu \text{ and } \lim_{n\to\infty} StdDev\left[z_n\right] = \sigma \tag{9}$$

However, this equilibrium state might never even practically occur, when considered that the stability condition does not denote a specific number of iterations to attain this state. Likewise, the particle velocities might also explode to very high magnitudes throughout the search process before achieving the equilibrium state.

III. Experimental Procedure

The purpose of the investigation is to empirically determine whether velocity explosion occurs, while CPs are sampled from the convergent region of Poli's stability condition. To this end, evaluation metrics are defined to assess the stability of the PSO search. This section thus provides an overview of the experimental procedure, focusing on evaluation metrics and specifics regarding the PSO implementation.

A. Evaluation Metrics

The metrics used to assess the PSO search behaviour are:

1) **Percentage of particles in infeasible space.** A particle resides in infeasible space if at least one of its dimensions exceeds the feasible search space boundaries.

2) **Percentage of particles that are stable.** Equation (8) in Section II-C gives Poli's stability condition, with the convergent region shown in Figure 1. If CPs are sampled from within the convergent region, they are considered stable.

3) **Average particle velocity** refers to the average step sizes taken by the particles. Velocities must decrease if the search process is to converge, but should not decrease so quickly that the search becomes stuck in a local minimum. Average particle velocity is calculated using [19]:

$$\Delta(t+1) = \frac{1}{n_s} \sum_{i=1}^{n_s} \|\mathbf{x}_i(t+1) - \mathbf{x}_i(t)\| \tag{10}$$

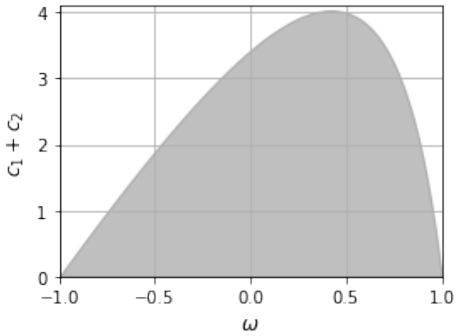

Fig. 1. Poli's stability condition convergent region [10], [22].

4) **Average swarm diversity** can give insight about the balance of exploration and exploitation [23], as it measures how spread out the particles are. Diversity is calculated [24] as

$$\mathcal{D} = \frac{1}{n_s} \sum_{i=1}^{n_s} \sqrt{\sum_{j=1}^{n_x} \left(x_{ij} - \bar{x}_j\right)^2} \tag{11}$$

where the swarm center is located at

$$x_j = \frac{\sum_{i=1}^{n_s} x_{ij}}{n_s} \tag{12}$$

where n_s is the number of particles in the swarm, and n_x is the number of dimensions.

B. Implementation

The PSO algorithm used is the inertia weight PSO, where the neighbourhood of each particle is the entire swarm [18], with governing equations given by Equations (1) and (2). The PSO is run on the functions given in Appendix A, stepping through different combinations of CPs, all of which are sampled from within the convergent region of Poli's stability condition. The inertia weight ω is stepped from 0 to 1 in increments of 0.025, while the acceleration coefficients c_1 and c_2 are stepped (seperately) from 0 to 4 in increments of 0.05, so as to create a 3D CP search space. The CP stepping results in a total of 13073 CP configurations. CP areas which result in large particle velocities are then further investigated by stepping the CPs in smaller increments of 0.01. Given the large number of PSO runs which are performed, the computational budget is quite high. Furthermore, since many runs have CP configurations which are extremely similar, this study does not perform a number of independent runs for every CP configuration.

Function dimensionality is set to $n_d = 30$, and the number of particles is also chosen as $n_s = 30$, with a maximum

of $i_{max} = 5000$ iterations for each run. Particles are initialized uniformly over all dimensions within the feasible search space. The objective function value is only computed for feasible particles, by setting infinite objective function values for particles in infeasible space. Doing so under the assumption of minimization automatically disqualifies those particles from global best position updates.

IV. Results

This section presents the results of the experimental procedure described in Section III, with figures to illustrate the search characteristics.

Figures 2 through 5 show PSO search behaviour plots for the metrics in Section III-A. Each figure shows the results for a different function for the specific CPs of $\omega = 0.72, c_1 = 3.37, c_2 = 0.01$. This CP configuration is plotted as it resulted in the largest velocities amongst those tested, but any number of other configurations also lead to massively exploded velocities.

Note that while the discussion will focus on Figure 2, the same discussion applies to all other functions, which are given here to demonstrate the similarity in search behaviour despite different landscapes. This indicates that velocity explosion is in larger part due to certain (stable) CPs, more so than due to the objective function landscape. The landscape of course still plays a role, seen therein that the velocities explode to different orders of magnitude in Figures 2c through 5c.

Figure 2a plots the percentage of stable particles and confirms all particles as stable, so to leave no doubt about the cause of the velocity explosion. Nonetheless, almost all particles are nigh permanently outside of the feasible space, as shown by the percentage of infeasible particles plot in Figure 2b. In principle, this is not a problem for the search process, and it has been shown that particles generally leave the feasible search space within the first few iterations [25]. However, if nearly the entire swarm is permanently in infeasible space, as is the case here, the effectiveness of the search process becomes questionable.

Figure 2c portrays how the average particle velocity explodes to above 2.0×10^{27} (and up to 1.0×10^{40} for Figure 5c!) Figure 2d portrays the explosion on the log scale, and clearly demonstrates that the particles do not, in fact, come back down to the order of magnitude of the search space, but remain exploded.

Velocity explosion naturally affects the swarm diversity calculation similarly, shown in Figure 2e. This is an additional problem with velocity explosion, because even if one particle assumes such a large velocity, it will dominate the swarm diversity calculation. The swarm will appear to be very diverse, even though it is not. The swarm diversity plot will also become equivalent to the particle velocity plot, because the highest velocity particles dominate in Equation (11). The latter can be seen clearly when Figures 2c and 2e are compared.

A 3D heatmap (log-scale) is given in Figure 6 to visualize where in CP space velocity explosion is most likely. The heatmap makes use of the maximum velocity attained during the search, and not the final velocity, as the choice of maximum time steps used is essentially arbitrary. However, as Figures 2c through 5c show, the maximum and final velocities are not much different, and it is unlikely to assume that the velocities would die out at any arbitrary, but fixed point in time. Despite the cautionary note against purely relying on Poli's criterion, Figure 6 actually also validates the accuracy of Poli's criterion, because the 3D region heatmap shows that the largest velocities are indeed attained close to the boundaries of the convergent region.

While stepping through the CP configurations, 26.5% of CP configurations resulted in particle velocities in excess of $\Delta_{lim} = 10^3$. Δ_{lim} was chosen to be an order of magnitude larger than the boundaries of the search space, which are 5×10^2 for the biggest landscape used. Similarly, 3.1% of CP configurations resulted in particle velocities in excess of 10^5, arbitrarily chosen to illustrate the extreme cases, with some particle velocities reaching up to 10^{40}. Ideally, particle velocities should remain in the order of magnitude of the search space, namely 10^2 for the functions as used here.

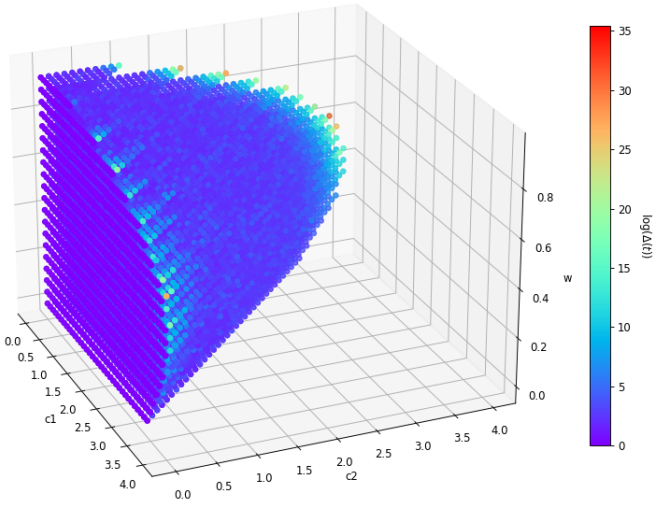

Fig. 6. 3D heatmap of maximum particle velocities (log) for all CP configurations.

V. Conclusion

An investigation of particle swarm optimization (PSO) search behaviour, constrained by Poli's stability condition, was presented in this study. The results illuminate challenges for self-adaptive PSO (SAPSO) strategies which attempt to dynamically adjust the control parameter (CP) configuration, as well as for the use of stability conditions in PSO more generally. Despite the theoretical promise of stability conditions, the observed practical behavior often demonstrates considerable divergence of particles due to the intractability of running the PSO search to the mathematical limit.

Velocity explosion was empirically demonstrated, with particle velocities of up to 10^{40} observed during the search process. The study unambiguously demonstrated the discrepancy between the theoretical promise of stability conditions,

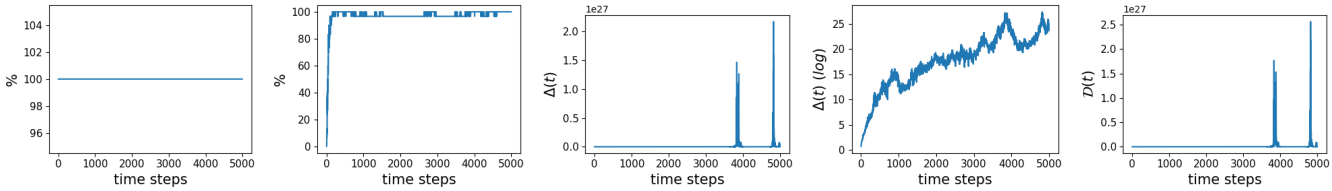

Fig. 2. Ackley1: (a) Stable particles (b) Infeasible particles (c) Particle velocities (d) Particle velocities (log) (e) Swarm diversity

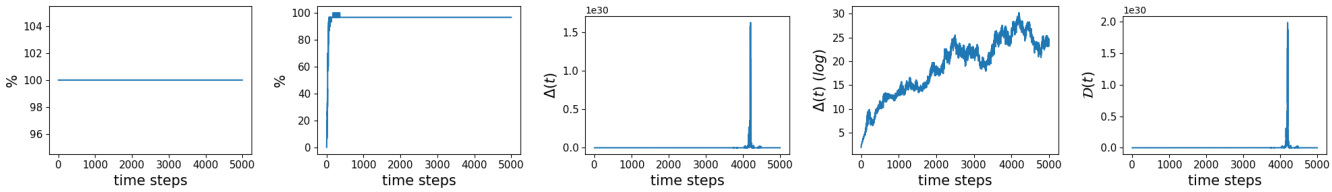

Fig. 3. Rana: (a) Stable particles (b) Infeasible particles (c) Particle velocities (d) Particle velocities (log) (e) Swarm diversity

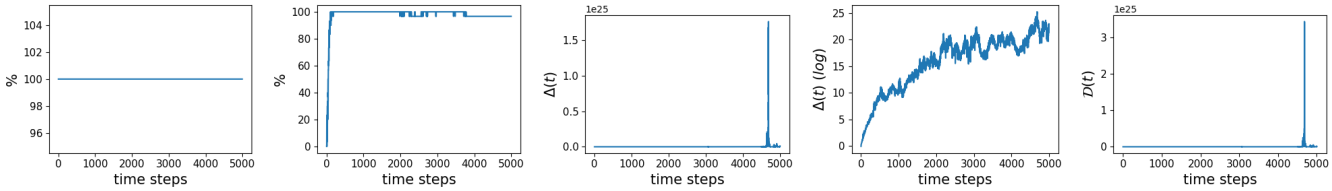

Fig. 4. Rastrigin: (a) Stable particles (b) Infeasible particles (c) Particle velocities (d) Particle velocities (log) (e) Swarm diversity

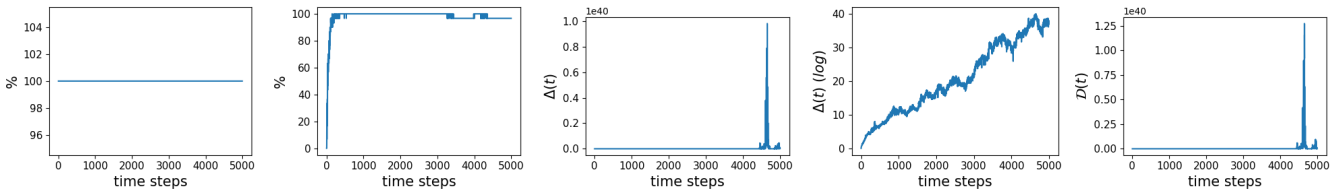

Fig. 5. XinSheYang1: (a) Stable particles (b) Infeasible particles (c) Particle velocities (d) Particle velocities (log) (e) Swarm diversity

and their real-world performance. It therefore emphasizes caution against relying solely on Poli's criterion or similar theoretical constructs to guarantee a stable search process.

Nonetheless, the empirical results confirm Poli's stability condition to be accurate, since velocity explosion occurs with increasing frequency as CPs approach the stability boundary.

Moreover, the research establishes the pressing need for a deeper understanding of the practical implications of stability conditions in PSO and CP tuning. The criticality of empirical evidence in the evaluation of swarm intelligence algorithms performance is underscored.

REFERENCES

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

[2] T. Beielstein, K. E. Parsopoulos, and M. N. Vrahatis, "Tuning PSO parameters through sensitivity analysis," Universitätsbibliothek Dortmund, Technical Report Interner Bericht des Sonderforschungsbereichs (SFB) 531 Computational Intelligence No.CI-124/02, 2002.

[3] F. van den Bergh and A. Engelbrecht, "A study of particle swarm optimization particle trajectories," *Information Sciences*, vol. 176, no. 8, pp. 937–971, 2006. [Online]. Available: https://doi.org/10.1016/j.ins.2005.02.003

[4] M. R. Bonyadi and Z. Michalewicz, "Impacts of coefficients on movement patterns in the particle swarm optimization algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 378–390, 2016.

[5] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," in *Proceedings of the IEEE Swarm Intelligence Symposium*. IEEE, 2007, pp. 120–127.

[6] D. Wolpert and W. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[7] M. Jiang, Y. Luo, and S. Yang, "Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm," *Information Processing Letters*, vol. 102, no. 1, pp. 8–16, 2007.

[8] V. Kadirkamanathan, K. Selvarajah, and P. Fleming, "Stability analysis of the particle dynamics in particle swarm optimizer," *IEEE Transac-*

tions on Evolutionary Computation, vol. 10, no. 3, pp. 245–255, 2006.
[9] V. Gazi, "Stochastic stability analysis of the particle dynamics in the pso algorithm," in *Proceedings of the IEEE International Symposium on Intelligent Control*. Dubrovnik, Croatia: IEEE Press, 2012, pp. 708–713.
[10] R. Poli, "Mean and variance of the sampling distribution of particle swarm optimizers during stagnation," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 712–721, 2009.
[11] C. Cleghorn and A. Engelbrecht, "A generalized theoretical deterministic particle swarm model," *Swarm Intelligence*, vol. 8, pp. 35–59, 2014.
[12] C. W. Cleghorn and A. P. Engelbrecht, "Particle swarm stability: A theoretical extension using the non-stagnate distribution assumption," *Swarm Intelligence*, vol. 12, pp. 1–22, 2018.
[13] C. W. Cleghorn and A. Engelbrecht, "Unified particle swarm optimizer: Convergence analysis," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2016, pp. 447–454.
[14] C. W. Cleghorn and A. P. Engelbrecht, "Fully informed particle swarm optimizer: Convergence analysis," in *Proceedings of the IEEE Congress on Evolutionary Computation*. Piscataway, NJ: IEEE Press, 2015, pp. 164–170.
[15] C. Cleghorn and A. Engelbrecht, "Particle swarm variants: standardized convergence analysis," *Swarm Intelligence*, vol. 9, 09 2015.
[16] C. W. Cleghorn and A. P. Engelbrecht, "Particle swarm convergence: An empirical investigation," in *2014 IEEE Congress on Evolutionary Computation (CEC)*, Beijing, China, July 2014, pp. 2524–2530.
[17] ——, "Particle swarm variants: standardized convergence analysis," *Swarm Intelligence*, vol. 9, pp. 177–203, 2015.
[18] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, vol. 6, 06 1998, pp. 69–73.
[19] K. Harrison, A. P. Engelbrecht, and B. Ombuki-Berman, "Self-adaptive particle swarm optimization: a review and analysis of convergence," *Swarm Intelligence*, vol. 12, pp. 187–226, 09 2018.
[20] K. R. Harrison, A. P. Engelbrecht, and B. M. Ombuki-Berman, "Optimal parameter regions and the time-dependence of control parameter values for the particle swarm optimization algorithm," *Swarm and Evolutionary Computation*, vol. 41, pp. 20–35, 2018.
[21] ——, "The sad state of self-adaptive particle swarm optimizers," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2016, pp. 431–439.
[22] R. Poli and D. Broomhead, "Exact analysis of the sampling distribution for the canonical particle swarm optimiser and its convergence during stagnation," in *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: Association for Computing Machinery, 2007, pp. 134–141.
[23] P. Bosman and A. P. Engelbrecht, "Diversity rate of change measurement for particle swarm optimisers," in *Swarm Intelligence*, M. Dorigo, M. Birattari, S. Garnier, H. Hamann, M. Montes de Oca, C. Solnon, and T. Stützle, Eds. Cham: Springer International Publishing, 2014, pp. 86–97.
[24] O. Olorunda and A. P. Engelbrecht, "Measuring exploration/exploitation in particle swarms using swarm diversity," in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, 2008, pp. 1128–1134.
[25] S. Helwig and R. Wanka, "Theoretical analysis of initial particle swarm behavior," in *Proceedings of the Tenth International Conference on Parallel Problem Solving from Nature*, 2008, pp. 889–898.
[26] A. Gavana, "Global Optimization Benchmarks and AMPGO," 2022. [Online]. Available: www.infinity77.net/global_optimization/genindex.html

APPENDIX A
FUNCTIONS

The functions used are given in below, with their properties in Table I.

- Ackley 1:

$$f(\mathbf{x}) = -20e^{-0.2\sqrt{\frac{1}{n}\sum_{j=1}^{n} x_i^2}} - e^{\frac{1}{n}\sum_{j=1}^{n}\cos(2\pi x_i)} + 20 + e \tag{13}$$

for $x_i \in [-32, 32]$

- Rana:

$$f(\mathbf{x}) = \sum_{i=1}^{n-1}(x_{i+1}+1)\cos(t_2)\sin(t_1) + x_i\cos(t_1)\sin(t_2) \tag{14}$$

where

$$t_1 = \sqrt{|x_{i+1} + x_i + 1|}$$

and

$$t_2 = \sqrt{|x_{i+1} - x_i + 1|}$$

for $x_i \in [-500, 500]$

- Rastrigin:

$$f(\mathbf{x}) = 10n + \sum_{i=1}^{n}\left(x_i^2 - 10\cos(2\pi x_i)\right) \tag{15}$$

for $x_i \in [-5.12, 5.12]$

- XinSheYang 1:

$$f(\mathbf{x}) = \sum_{i=1}^{n}\epsilon_i |x_i|^i \tag{16}$$

for $\epsilon_i \sim U(0, 1)$

TABLE I
FUNCTION CHARACTERISTICS [26] (C=CONTINUOUS, NC=NON-CONTINUOUS, D=DIFFERENTIABLE, ND=NON-DIFFERENTIABLE, S=SEPARABLE, NS=NON-SEPARABLE, MM=MULTI-MODAL, UM=UNIMODAL.)

Function	Eq.	Cont.	Diff.	Sep.	Mod.
Ackley 1	13	C	D	NS	MM
Rana	14	C	ND	NS	MM
Rastrigin	15	C	D	S	MM
XinSheYang 1	16	NC (?)	D (?)	S	MM