Particle Swarm Optimisation: An Analysis of Particle Attactors

BS Steyn
(Department of Computer Science)
Stellenbosch University
Stellenbosch, South Africa
21740178@sun.ac.za

***Abstract*—Different attractors are a common technique used to alter the ratio between exploration and exploitation. This paper examines the performance of particle swarm optimisation when using variations of the standard inertia-weight pso. The variations are compared to the standard inertia-weighted PSO. The study concluded that the variation with the correct approach can perform better on average than the standard inertia weight PSO**

I. Introduction

Particle swarm optimization (PSO) is a stochastic optimization algorithm modelled after the behaviour of birds in a flock which was established by Kennedy and Eberhart in 1995 [4]. PSO belongs to the larger field of swarm intelligence (SI). In PSO, a swarm of particles search for a candidate solution by iterative updates to the positions of particles. The search is guided by both the best local position a particle has found as well as the best position found by the neighbourhood particles, these are the standard attractors. The quality of a specific solution is evaluated by means of an objective function of a predetermined optimization problem. PSO performance is highly dependent on the selection of appropriate control parameters (CP). Inappropriate choices for these parameters may cause particles to exhibit roaming behaviour [2]. It has been shown both empirically and theoretically that if control parameters are selected to satisfy certain conditions, the particles eventually converge innately [1] [5] [6] [7]. The contribution of this paper lies in the exploration of adding an additional attractor, the attractor added is the best position from the swarm of the specific iteration. This paper will evaluate whether the new attractor is beneficial towards the swarm's exploration vs exploration behaviour.

II. Background

This section elaborates on the PSO algorithm, the control parameter configurations and the topology of the swarm.

A. Particle swarm optimization algorithm

Since the initial contributions of Kennedy and Eberhart various PSO techniques have been developed. This paper makes use of the inertia weight model of PSO, developed by Shi and Eberhart [4]. This paper also explores the variation of the inertia weight model of PSO, wherein an addition attractor is used. Both the inertia weight model of PSO and our variation establish a proportional relationship between a particle's velocity at the previous iteration and the particle's velocity at the current iteration. According to the inertia weight model of PSO, the equations to update the velocity and position of a particle are given by

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1} \tag{1}$$

$$\mathbf{v}_i^{t+1} = w\mathbf{v}_i^t + c_1\mathbf{r}_{1i}(\mathbf{y}_i^t - \mathbf{x}_i^t) + c_2\mathbf{r}_{2i}(\hat{\mathbf{y}}_i^t - \mathbf{x}_i^t) \tag{2}$$

and for the variation in this paper, the equation to update the position of a particle is given by (1) while the equation to update velocity is given by

$$\mathbf{v}_i^{t+1} = w\mathbf{v}_i^t + c_1\mathbf{r}_{1i}(\mathbf{y}_i^t - \mathbf{x}_i^t) + c_2\mathbf{r}_{2i}((1-\lambda)(\hat{\mathbf{y}}_i^t - \mathbf{x}_i^t) + \lambda(\bar{\mathbf{y}}_i^t - \mathbf{x}_\mathbf{i}^\mathbf{t})) \tag{3}$$

Bold symbols in the equations above represent n dimensional vectors that correspond to the dimensionality of the search domain. The symbols of equations (1), (2) and (3) are clarified below:

- $\mathbf{x}_i^t$ represent the the position of i^th particle at iteration t. The initial positions of the particles are randomly allocated within the search space.
- $\mathbf{x}_i^{t+1}$ represent the the position of i^{th} particle at iteration $t+1$.
- $\mathbf{v}_i^t$ represents the velocity of the i^{th} particle at iteration t. The initial velocity of all particles in the swarm are initialised to the zero vector.
- $\mathbf{v}_i^{t+1}$ represents the velocity of the i^{th} particle at iteration $t+1$.
- $\mathbf{y}_i^t$ represents the position of the best function evaluation found by the i^{th} particle, defined as the personal best position. This position is updated if the i^{th} particle personally finds a position with a better function evaluation.
- $\hat{\mathbf{y}}_i^t$ represents the position of the best function evaluation found by the neighbourhood the i^{th} particle belongs to,

defined as the neighbourhood best position. This position is updated if any particle belonging to the neighbourhood finds a better function evaluation.

- $\bar{\mathbf{y}}_i^t$ represents the position of the best function evaluation in the current iteration, defined as the iteration-best position. This position updates every iteration.
- w represents the inertia coefficient. w determines the effect the previous velocity of the i^{th} particle has on the subsequent iteration.
- c_1 represents the cognitive coefficients. c_1 determines the effect the personal best position has on the iteration.
- c_2 represents the social coefficients. c_2 determines the effect the neighbourhood best position has on the iteration.
- $\mathbf{r}_{2i}$ and $\mathbf{r}_{1i}$ introduce stochasticity into the model. $\mathbf{r}_{2i}$ and $\mathbf{r}_{1i}$ are sampled from a uniform distribution over (0,1).
- λ is used to balance the two social attractors, the neighbourhood best and the iteration best.

The algorithm used in this paper is clearly illustrated in Algorithm 1.

Algorithm 1 Standard PSO Algorithm

Create and initialize an n_x-dimensional swarm, S;
repeat
  for each particle i=1,...,$S.n_s$ **do**
    if $f(S.x_i) < f(S.y_i)$ **then**
      $S.y_i = S.x_i$;
    end if
    if $f(S.y_i) < f(S.\hat{y}_i)$ **then**
      $S.\hat{y}_i = S.y_i$;
    end if
  end for
  for each particle i=1,...,$S.n_s$ **do**
    update the velocity;
    update the position;
  end for
until stopping condition is true

B. Control parameter confgurations

A PSO particle is considered stable if it has control parameters w, c_1, c_2 that satisfy the following equation. This has been both theoretically and empirically proven that the particle will reach an equilibrium state while adhering to

$$c_1 + c_2 < \frac{24(1 - w^2)}{7 - 5w} \text{ for } w \in [-1, 1] \tag{4}$$

All the control parameters used in this paper will adhere to (4).

C. Exploration vs Exploitation

The balance between exploration and exploitation is an important factor in the success of PSO [8]. A PSO implementation should focus on exploration early on, then eventually start to focus on the exploitation of promising solutions. *Swarm diversity* is a useful metric to quantify the exploratory tendencies of a swarm. Swarm diversity refers to the average Euclidean distance between a particle, and the position of the average particle. The higher the swarm diversity the further the particles are from each other, while lower swarm diversity implies the swarm is converging.

D. Communication Topology techniques

The social influence that particles have on each other is highly dependent on the chosen communication topology of the swarm. The chosen communication topology determines the speed at which good solutions are spread between different neighbourhoods [7]. The most commonly used are the *ring* and *star* topologies [2]. This paper will use the star topology in which there is only one global neighbourhood.

III. Evaluation Metrics

The stochastic parameters $\mathbf{r}_1$ and $\mathbf{r}_2$ leads to PSO being a non-deterministic process. The same starting configuration may lead to different results. To counter the non-deterministic nature of PSO the simulation will be run 30 times. The results of each simulation will be averaged as the performance over time is of interest. The performance metrics used in this paper are listed below:

1) Global best position
2) Swarm Diversity
3) Average velocity magnitude of the swarm

The merits of the metrics are given below:

- The *Global best position* represents the best solution the swarm finds. The merit of this metric is to have a clear metric to track whether the swarm can solve the problem.
- *Swarm diversity* is the metric responsible for tracking the exploratory characteristic of the swarm. When diversity is high, particles are exploring the search space whereas when diversity is low the particles are exploiting The merit of this metric is that the relationship between exploration and exploitation is a very important characteristic of PSO. Particles must focus on exploring in the early stages but eventually switch to exploitation in the later stages.
- *Average velocity magnitude* represents the intensity of acceleration that particles within the swarm are exposed to on average. The merit of this metric is that it provides insight into the granularity of the search process.

The above metrics are evaluated at every iteration and averaged across all the iterations. For all simulations, a swarm of 30 particles will traverse a 30-dimensional search space for 5000 iterations. The intention is to gather data on the effect of the variations on the inertia weight PSO, proposed in this paper.

IV. Simulation Procedure

This section highlights the empirical process used to compare the performance of the variations on inertia weight PSO. To determine how the variation performs under a diverse set of conditions a variety of benchmark functions have been selected. These functions have been selected as they provide diverse spread over uni-modal, multi-modal, separable and non-separable. function.

A. Selected Benchmark functions

Five benchmark functions are used to evaluate the performance of PSO implementation. The benchmark functions selected have a global minimum at the point $x^* = (0, 0, ..., 0)$.

1) $Ackely\ 1\ Function$:

$$f(\mathbf{x}) = -20 \exp^{-02\sqrt{\frac{1}{n}\sum_{j=1}^{n} x_i^2}} - \exp^{\frac{1}{n}\sum_{j=i^n} \cos(2\pi x_i)} + 20 + \exp \quad where\ -32 \leq x_i \leq 32 \tag{5}$$

2) $Brown\ Function$:

$$f(\mathbf{x}) = \sum_{i=1}^{n-1} (x_i^2)^{(x_{i+1}^2+1)} + (x_{i+1}^2)^{(x_i^2+1)} \quad where\ -1 \leq x_i \leq 1 \tag{6}$$

3) $Alpine\ 1\ Function$:

$$f(\mathbf{x}) = \sum_{i=1}^{n} |x_i \sin(x_i) + 0.1 x_i| \ where\ -20 \leq x_i \leq 20 \tag{7}$$

4) $Discus\ Function$:

$$f(\mathbf{x}) = 10^6 x_1^2 + \sum_{i=2}^{n} x_i^2 \ where\ -100 \leq x_i \leq 100 \tag{8}$$

5) $Drop\ wave\ Function$:

$$f(\mathbf{x}) = \frac{1 + \cos(12\sqrt{\sum_{i=1}^{n} x_i^2})}{2 + 0.5\sum_{i=1}^{n} x_i^2} \quad where\ -5.12 \leq x_i \leq 5.12 \tag{9}$$

In the above equation, n is defined as the number of dimensions, and i is defined as a particular dimension. The global best particle was updated synchronously as suggested in the original literature [10]. A synchronous global update strategy performs neighborhood best updates separately from position updates. This leads to the changes in the global best only being integrated into the model in the following iteration [3]. The iteration best solution is also updated synchronously in this paper's variation.

B. Variation parameters

Various strategies are used to determine an optimal approach to setting our λ parameter. The strategies are

- The first approach is to tune the parameter λ manually.
- The second approach is to randomly sample from a $uniform(0, 1)$ distribution.
- The third approach is to use a dynamically changing value starting at $\lambda(0) = 1$, linearly decreased to $\lambda(n_t) = 0$, n_t is the maximum number of iterations.
- The fourth approach is to let λ be a vector of random values in [0,1]. so that each dimension has a different balance between the two social attractors.

C. Simulation parameters

In this paper a different set of control parameters will be used for each function as these parameters were found to have the best outcome on their respective function. The different values for the control parameters can be seen in the table below.

Function	w	c_1	c_2
Ackley 1	0.6	2.0	1.8
Alpine 1	0.65	1.6	1.8
Brown	0.6	1.8	1.6
Discus	0.75	2.0	1.0
Drop Wave	0.8	2.0	1.6

The performance of the variations on the benchmark functions (5) (6) (7) (8) (9) will be evaluated using the metrics mentioned in the previous section.

V. Simulation results

This section presents the empirical results and observations of how the variations performed on the various benchmark functions compared to the standard inertia weight PSO.

A. Performance of standard inertia PSO

To establish a baseline the performance of the standard inertia weight PSO will be evaluated on the benchmark functions.

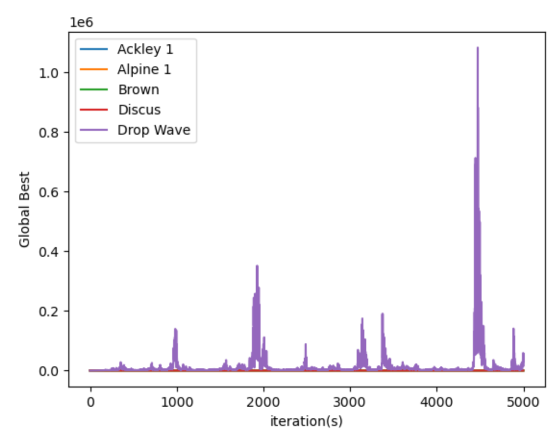


Fig. 1: Swarm diversity of standard inertia weight PSO.

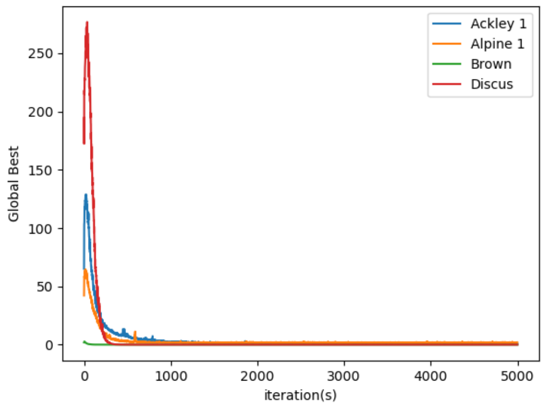


Fig. 2: Swarm diversity of standard inertia weight PSO without the $drop\ wave\ function$.

Figure 1 highlights that the $drop\ wave\ function$ does not exhibit the explore then exploit nature we would like to see in a PSO. This is likely due to the extreme local minimum found in the $drop\ wave\ function$ these minimums are extreme enough to cause particles to get stuck. Figure 2 on the other hand demonstrates that the inertia weight PSO does exhibit the explore then exploit nature that we would expect from the algorithm.

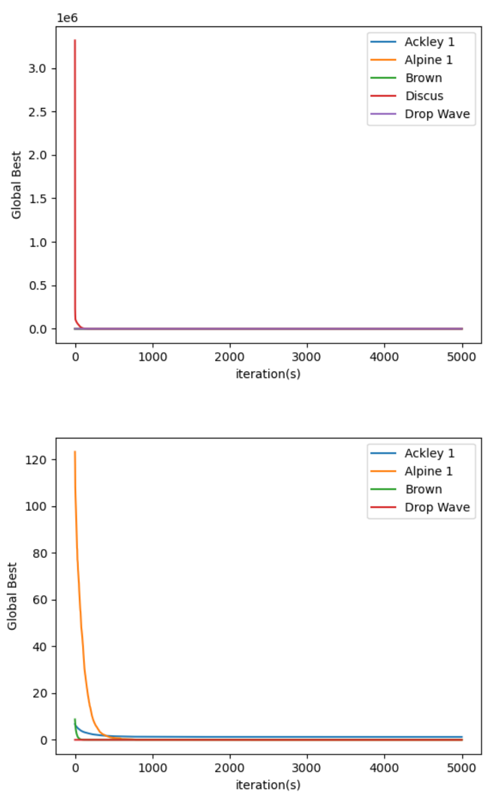
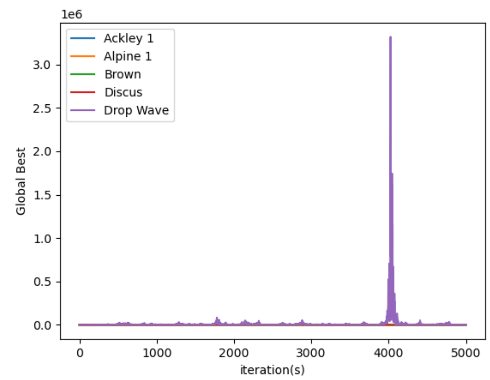


Fig. 3: Mean Global best position for inertia weight PSO

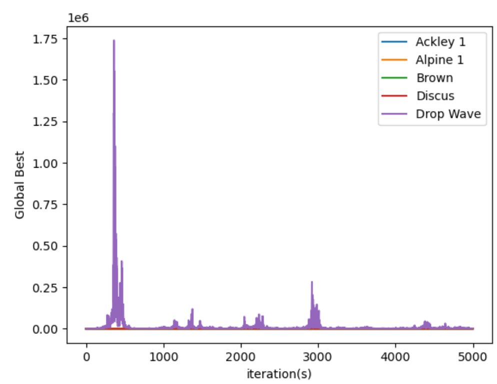
Figure 3 highlights that the inertia weight PSO does converge to a global minimum even for the $drop\ wave\ function$. The mean velocity is similarly distributed to the swarm diversity in this case. It can thus be seen that the standard inertia weight PSO performed well in most cases but can become stuck if the local minima are extreme enough as shown in Figure 1.

B. Performance of the variations

The behaviour of the variations is similar to the standard inertia weight PSO. However it can be observed that there is an improvement in the $drop\ wave\ function$ for some of the variations.



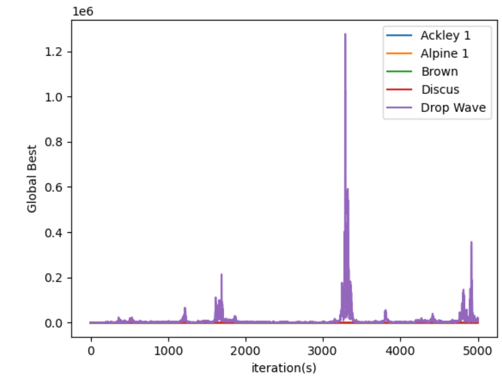
(a) First approach



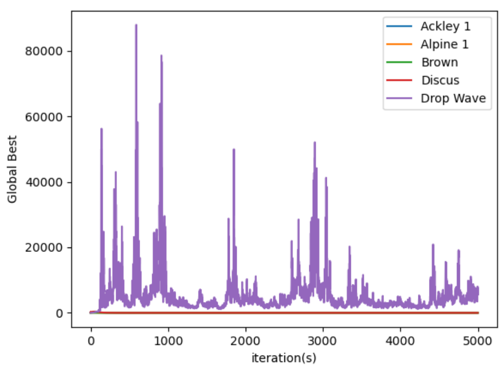
(b) Second approach

Fig. 4: Swarm diversity for the first two variations

Figure 4 highlights that the first and second approaches have the same problem concerning swarm diversity. However, in both cases, there is an improvement as it is highly concentrated, as seen in Figure 4. This is to be expected as the variations introduce the iteration-best attractor which causes particles to explore more while converging. Allowing the particles to escape more local minima. In Figure 4 (b) it is clear that the added stochastic element to the λ value helps to uphold the desired explore then exploit nature. This makes sense as the added random element helps particles escape local minimums.



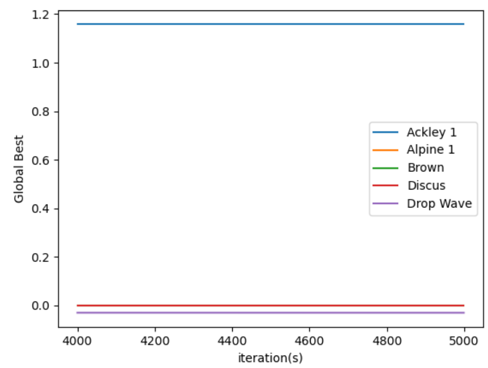
(a) Third approach



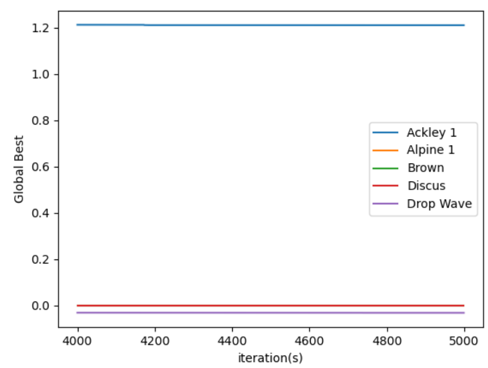
(b) Fourth approach

Fig. 5: Swarm diversity for the last two variations

Figure 5 (a) highlights that without the stochastic element, the variation performs similarly to the standard inertia weight PSO. Figure 5 (b) once again shows that the stochastic element is important to achieve the explore then exploit nature that is desired. If you compare Figure 5 (b) to Figure 4 (a) it highlights that by adding the stochastic element to each dimension separately the algorithm is slower to converge. Figure 5 (b) highlights that adding the stochastic element to each dimension provides a better balance between exploration and exploitation. Figure 4 (a) highlights a quick swap between exploration and exploitation.
Figure 6 shows the global best found by the standard inertia weight PSO algorithm and the fourth approach variation in the last 1000 iterations. The Figure shows that the difference in the quality of solutions found does not vary by a large degree. This holds true for all the variation approaches.



(a) Fourth approach



(b) Standard Inertia weight PSO

Fig. 6: Swarm diversity for the last two variations

C. *Rankings of Global Solutions Found*

In the table below it shows which approach worked best for which benchmark function and which performed best overall.

Function	Inertia weight PSO	1st Approach	2nd Approach	3rd Approach	4th Approach
Ackley 1	5	1	2	4	3
Alpine 1	5	4	1	3	2
Brown	1	5	4	3	2
Discus	4	1	3	2	5
Drop Wave	1	4	2	5	3
Average	3.2	3	2.4	3.4	3

VI. Conclusion

The intention of this paper was to investigate the effect of an additional social attractor. As well as the best balance between the classic social attractor and the additional social attractor. In this paper, it was shown that using the variation with the 2nd approach yielded the best result, while the other approaches performed similarly to the standard inertia weight PSO. Metrics were established to determine the quality leading to the results above. Simulations were conducted and based on the results, determined by the evaluation metrics it was found that the variation of the social attractor did not have a great effect on the performance in terms of finding the optimal global solution. With the exception of the second approach.
Future work includes a study to further tune the λ parameter, applying the approaches to a dynamic space and determining if

the rankings hold for different values of the iteration, number of particles and dimensions.

REFERENCES

[1] Christopher Cleghorn and Andries Engelbrecht. "Particle swarm convergence: An empirical investigation". In: July 2014. DOI: 10.1109/CEC.2014.6900439.

[2] 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. DOI: 10.1109/MHS.1995.494215.

[3] A. P. Engelbrecht. "Roaming Behavior of Unconstrained Particles". In: *Proceedings of the 2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence*. BRICS-CCI-CBIC '13. USA: IEEE Computer Society, 2013, pp. 104–111. ISBN: 9781479931941. DOI: 10.1109/BRICS-CCI-CBIC.2013.28. URL: https://doi.org/10.1109/BRICS-CCI-CBIC.2013.28.

[4] Anshul Gopal, Mohammad Sultani, and Jagdish Bansal. "On Stability Analysis of Particle Swarm Optimization Algorithm". In: *Arabian Journal for Science and Engineering* 45 (July 2019). DOI: 10.1007/s13369-019-03991-8.

[5] J. Kennedy and R. Eberhart. "Particle swarm optimization". In: *Proceedings of ICNN'95 - International Conference on Neural Networks*. Vol. 4. 1995, 1942–1948 vol.4. DOI: 10.1109/ICNN.1995.488968.

[6] Ryan Dieter Lang and Andries Petrus Engelbrecht. "An Exploratory Landscape Analysis-Based Benchmark Suite". In: *Algorithms* 14.3 (2021). ISSN: 1999-4893. DOI: 10.3390/a14030078. URL: https://www.mdpi.com/1999-4893/14/3/78.

[7] Xiao-Fang Liu, Zhi-Hui Zhan, and Jun Zhang. "Incremental particle swarm optimization for large-scale dynamic optimization with changing variable interactions". In: *Appl. Soft Comput.* 141.C (July 2023). ISSN: 1568-4946. DOI: 10.1016/j.asoc.2023.110320. URL: https://doi.org/10.1016/j.asoc.2023.110320.

[8] Riccardo Poli. "Mean and Variance of the Sampling Distribution of Particle Swarm Optimizers During Stagnation". In: *Evolutionary Computation, IEEE Transactions on* 13 (Sept. 2009), pp. 712–721. DOI: 10.1109/TEVC.2008.2011744.

[9] Ming Xu and JiangPing Gu. "Parameter selection for particle swarm optimization based on Stochastic multi-objective optimization". In: Nov. 2015, pp. 2074–2079. DOI: 10.1109/CAC.2015.7382846.