

Particle Swarm Optimization for Large-Scale Optimization

*An evaluation of different swarm-based approaches to solving large-scale optimization problems

BS Steyn

(*Department of Computer Science*)

Stellenbosch University

Stellenbosch, South Africa

21740178@sun.ac.za

Abstract—This paper proposes a modified Particle Swarm Optimisation (PSO) algorithm for large-scale optimisation, combining Linearly Increasing Grouping and Subspace Initialisation. These techniques improve exploration by initially restricting the search to a subspace and progressively activating more dimensions during optimisation. Experimental results on 100-dimensional benchmark functions show improved convergence, swarm diversity, and stability compared to standard PSO.

I. INTRODUCTION

Large-scale optimisation problems, characterised by high-dimensional search spaces, pose significant challenges for traditional meta heuristic algorithms such as Particle Swarm Optimisation (PSO) [4]. As the number of dimensions increases, PSO often suffers from premature convergence, reduced swarm diversity, and ineffective exploration. Several techniques have been proposed to improve the scalability of PSO, including subspace-based methods and group-wise dimensional control.

This paper proposes a hybrid PSO variant that combines Linearly Increasing Grouping and Subspace Initialisation [6] to improve convergence behaviour on large-scale optimisation problems. Linearly Increasing Grouping regulates the number of active dimensions in the velocity update equation, gradually transitioning from a reduced subspace to the full-dimensional space. Subspace Initialisation limits the initial positions and velocities of particles to a randomly selected subregion of the search space, thereby focusing exploration in the early search stages. The aim of this integration is to maintain high swarm diversity and promote structured convergence across complex landscapes.

The performance of the proposed approach is evaluated using five high-dimensional benchmark functions. Metrics such as global best value per iteration, swarm diversity, and average velocity magnitude are used to assess the effectiveness of the modifications. Results demonstrate that the proposed method achieves improved convergence stability and better balance between exploration and exploitation compared to the standard PSO algorithm.

The remainder of this paper is structured as follows: Section

II provides a background for this paper. Section III provides the details on the implementation used in this paper. Section IV provides the evaluation criteria and method used in this paper. Section V provides the findings of this paper. Section VI concludes the paper.

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 [8], and follows the general convergence criteria and stability analysis described by Clerc [1]. 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

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

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

Bold symbols in the equations above represent n dimensional vectors that correspond to the dimensionality of the search domain. The symbols of equations (1) and (2) 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 $(0,1)$.

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

Algorithm 1 Standard PSO Algorithm

```

1: Create and initialize an  $n_x$ -dimensional swarm,  $S$ ;
2: repeat
3:   for each particle  $i=1,\dots,S.n_s$  do
4:     if  $f(S.x_i) < f(S.y_i)$  then
5:        $S.y_i = S.x_i$ ;
6:     end if
7:     if  $f(S.y_i) < f(S.\hat{y}_i)$  then
8:        $S.\hat{y}_i = S.y_i$ ;
9:     end if
10:    end for
11:    for each particle  $i=1,\dots,S.n_s$  do
12:      update the velocity;
13:      update the position;
14:    end for
15:  until stopping condition is true

```

B. Linearly Increasing Grouping

Linearly Increasing Grouping (LIG) [6] is a strategy designed to improve the scalability of the Particle Swarm Optimisation (PSO) algorithm when applied to large-scale optimisation problems. The technique divides the n -dimensional search space into g equally sized groups and incrementally increases the number of active groups throughout the optimisation process. This results in a progressive inclusion of dimensions during swarm updates.

The number of active groups at iteration t is given by

$$G(t) = \left\lceil G_{\max} \cdot \frac{t}{T} \right\rceil \quad (3)$$

where G_{\max} is the total number of groups and T is the maximum number of iterations. Only the first $G(t)$ groups are updated at each iteration, while the remaining groups remain static.

This modification reduces the dimensionality of the search early in the process, promoting exploration, and gradually increases the search space dimensionality to allow for fine-tuned exploitation. The incremental dimensionality control improves swarm diversity and reduces the likelihood of premature convergence. LIG is particularly effective when the optimisation landscape contains deceptive basins or when the curse of dimensionality limits the effectiveness of full-space updates in early iterations.

C. Subspace Initialisation

Subspace Initialisation [6] is a dimensionality reduction technique aimed at improving the initial exploration phase of Particle Swarm Optimisation (PSO) [4] when applied to large-scale optimisation problems. The method involves randomly selecting a lower-dimensional subspace of the full n -dimensional search space in which particles are initially distributed and evaluated.

Let $S \subset \mathbb{R}^n$ represent the selected subspace where $|S| < n$. Each particle's position is initially sampled within S , and velocity updates are restricted to the same subspace for a predefined number of iterations. After this exploratory phase, particles are projected back into the full n -dimensional space, where standard PSO updates resume.

The primary benefit of Subspace Initialisation is the focused exploration of a simplified landscape, which increases the probability of identifying promising regions early in the search. This approach reduces the effects of the curse of dimensionality during the initialisation phase, leading to improved convergence behaviour and swarm diversity. Subspace Initialisation is particularly beneficial in problems where high-dimensional spaces contain redundant or weakly influential dimensions.

D. Exploration vs Exploitation

The balance between exploration and exploitation is an important factor in the success of PSO [7]. 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.

E. 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 [5]. 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. IMPLEMENTATION

This section provides an overview of the implementation

A. Hybrid PSO

The proposed PSO algorithm in this paper combines Linearly Increasing Grouping and Subspace Initialisation to improve the algorithm's performance on large-scale optimisation problems. These modifications are designed to address the limitations of standard PSO, which include premature convergence, loss of swarm diversity, and ineffective exploration in high-dimensional search spaces. Subspace Initialisation focuses the swarm's early search within a reduced-dimensional region of the solution space, allowing the algorithm to explore promising directions before expanding to the full problem domain. In parallel, Linearly Increasing Grouping progressively introduces additional groups of dimensions into the optimisation process, thereby maintaining a structured exploration-exploitation balance as the search progresses. By integrating these two techniques, the modified PSO aims to delay convergence in the early stages, enhance the swarm's ability to escape local optima, and ultimately improve the reliability and scalability of the algorithm when solving complex, high-dimensional benchmark functions.

B. Control parameter configurations

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

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

IV. EXPERIMENTAL PROCEDURE

This section describes the experimental procedure followed.

A. Evaluation Metrics

The stochastic parameters r_1 and 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 per iteration
- 2) Swarm Diversity
- 3) Average velocity magnitude of the swarm

The merits of the metrics are given below:

- *Global best position per iteration* represents the best solution the swarm finds in each iteration. The merit of this metric is to have a clear metric to track whether the swarm can improve over time.
- *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 10 particles will traverse a 100-dimensional search space for 5000 iterations. The intention is to gather data on the effectiveness of PSO for large scale optimization.

B. Selected Benchmark functions

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

1) Ackely 1 Function:

$$f(\mathbf{x}) = -20 \exp^{-0.2\sqrt{\frac{1}{n}\sum_{j=1}^n x_j^2}} - \exp^{\frac{1}{n}\sum_{j=i^n} \cos(2\pi x_i)} + 20 + \exp \quad (5)$$

where $-32 \leq x_i \leq 32$

2) Brown Function:

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

where $-1 \leq x_i \leq 1$

3) Alpine 1 Function:

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

4) Discus Function:

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

5) Drop wave Function:

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

where $-5.12 \leq x_i \leq 5.12$

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 [4]. A synchronous global update strategy performs neighbourhood 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.

V. RESULTS

This subsection provides the comparative results of the experiments as well as a detailed discussion of the obtained results.

A. Standard PSO

This section provides the results for the standard PSO algorithm without modifications for the problems listed in section IV. The following figures provided the results for the evaluation metrics discussed in section IV.

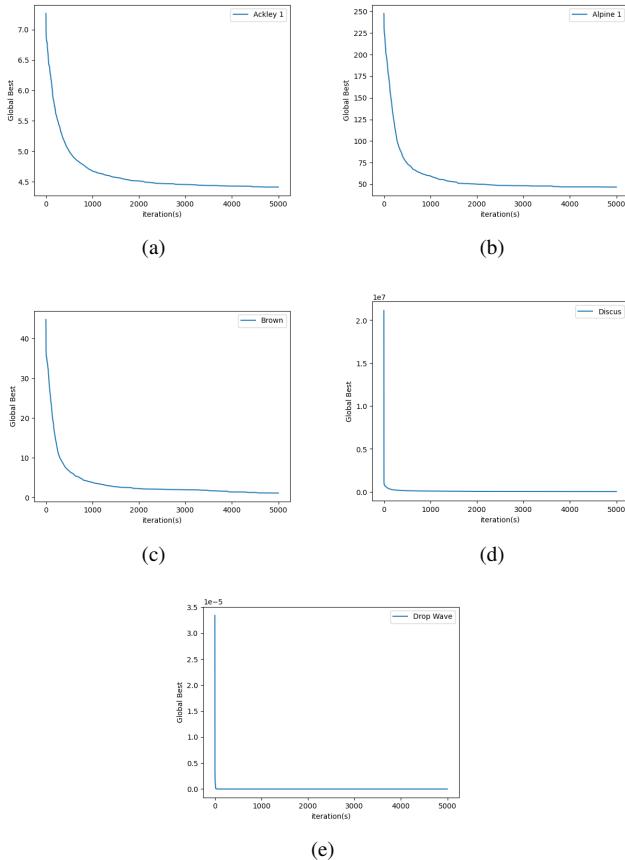


Fig. 1. (a) Global Best for Ackley 1 (b) Global Best for Alpine (c) Global best for Brown (d) Global best for Discus (e) Global best for Dropwave

Figure 1 illustrates the convergence behaviour of the standard PSO algorithm when applied to large-scale optimisation problems, with each benchmark function defined over 100 dimensions. For the Ackley 1, Alpine, and Brown functions, the algorithm demonstrates consistent improvement in global best values across iterations, suggesting that PSO retains adequate exploration capabilities under certain landscape conditions. In contrast, the Discus and Dropwave functions exhibit rapid early improvement followed by stagnation. The presence of ill-conditioning and deceptive basins in these functions leads to premature convergence and diminished swarm diversity. These observations emphasise the limitations of standard PSO in high-dimensional spaces and motivate the use of dimension-aware strategies such as linearly increasing group-based scal-

ing to enhance convergence reliability and exploratory behaviour.

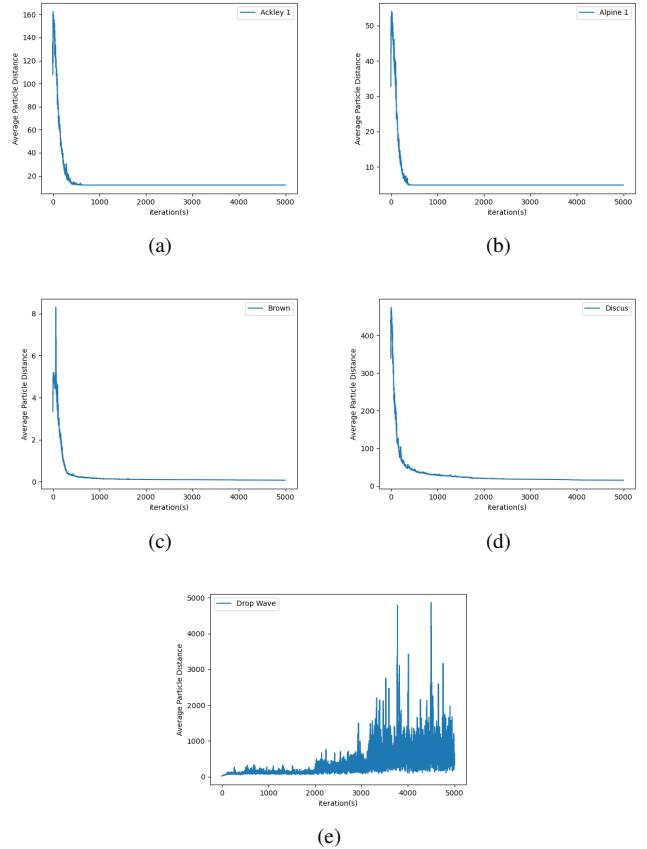
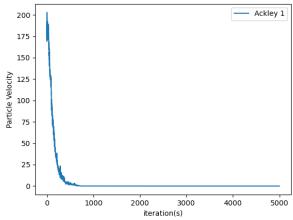
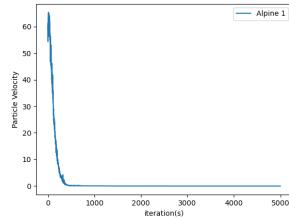


Fig. 2. (a) Average particle distance for Ackley 1 (b) Average particle distance for Alpine (c) Average particle distance for Brown (d) Average particle distance for Discus (e) Average particle distance for Dropwave

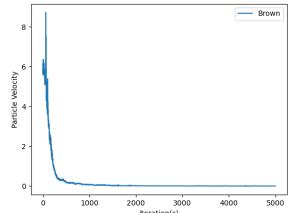
Figure 2 presents the swarm diversity of the standard PSO algorithm as measured by the average particle distance over 5000 iterations for benchmark functions defined over 100 dimensions. For the Ackley 1, Alpine, Brown, and Discus functions, a rapid decrease in diversity is observed, with the swarm quickly collapsing to a small region of the search space. This indicates a shift from exploration to exploitation within the first few hundred iterations, potentially resulting in premature convergence. In contrast, the Dropwave function demonstrates erratic and unstable diversity patterns, with frequent spikes in average particle distance. This behaviour suggests oscillatory search dynamics and the inability of the swarm to converge towards a stable region of the solution space. The instability can be attributed to the narrow and deceptive global minima characteristic of the Dropwave function. Overall, the results in Figure 2 highlight the sensitivity of standard PSO to landscape topology and emphasise the need for diversity-preserving mechanisms when addressing large-scale optimisation problems.



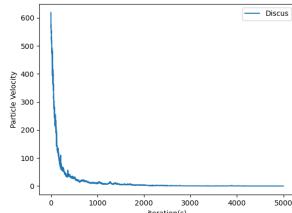
(a)



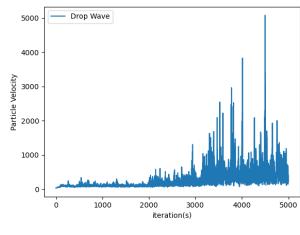
(b)



(c)



(d)



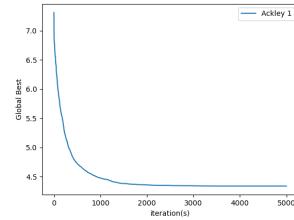
(e)

Fig. 3. (a) Average velocity magnitude for Ackley 1 (b) Average velocity magnitude for Alpine (c) Average velocity magnitude for Brown (d) Average velocity magnitude for Discus (e) Average velocity magnitude for Dropwave

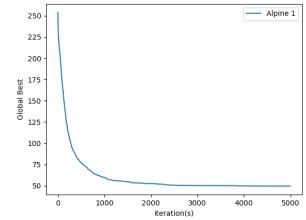
Figure 3 illustrates the average velocity magnitude of particles in the swarm for large-scale optimisation problems, each defined over 100 dimensions. For the Ackley 1, Alpine, Brown, and Discus functions, the particle velocities decrease sharply within the first few hundred iterations, after which the velocity stabilises near zero. This behaviour confirms that the swarm rapidly transitions from exploration to exploitation, and subsequently stagnates. The sharp decay in velocity suggests that the influence of cognitive and social components diminishes early in the search, limiting further positional updates. In contrast, the Dropwave function exhibits persistent velocity fluctuations throughout the optimisation process. The high and unstable velocity magnitudes observed in this case suggest that particles are unable to settle on a consistent search direction, which is indicative of poor convergence behaviour. These observations reinforce the claim that standard PSO is sensitive to landscape modality and may require adaptive velocity control mechanisms to maintain consistent progress in deceptive or rugged search spaces.

B. Linearly Increasing Grouping

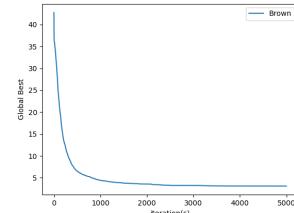
This section provides the results for the standard PSO algorithm with linearly increasing grouping for the problems listed in section IV. The following figures provided the results for the evaluation metrics discussed in section IV.



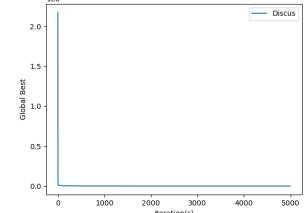
(a)



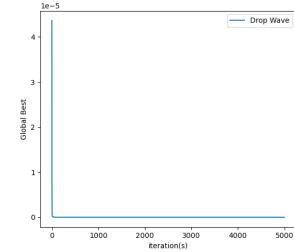
(b)



(c)



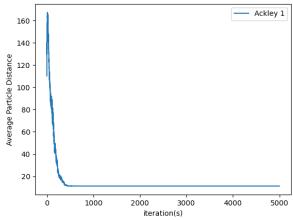
(d)



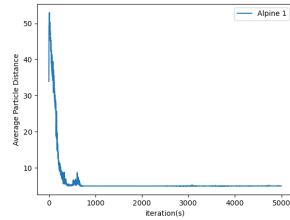
(e)

Fig. 4. (a) Global Best for Ackley 1 (b) Global Best for Alpine (c) Global best for Brown (d) Global best for Discus (e) Global best for Dropwave

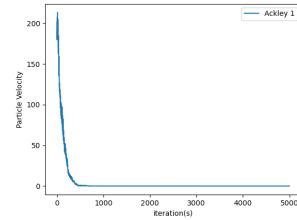
Figure 4 illustrates the convergence behaviour of the PSO algorithm modified with Linearly Increasing Grouping when applied to large-scale optimisation problems over 100 dimensions. For all benchmark functions, including Ackley 1, Alpine, and Brown, the algorithm maintains steady improvement in global best values, comparable to or slightly better than the standard PSO. However, for the Discus and Dropwave functions, the algorithm converges significantly faster, reaching the minimum function value within the first few hundred iterations. This accelerated convergence suggests that Linearly Increasing Grouping effectively guides the swarm towards optimal regions early in the search. While such rapid convergence may indicate improved exploitation, it may also suggest reduced exploration in later iterations, potentially risking entrapment in suboptimal regions for other problem types. These observations indicate that while Linearly Increasing Grouping enhances convergence speed, particularly on ill-conditioned or deceptive landscapes, careful calibration is required to maintain an appropriate balance between exploration and exploitation.



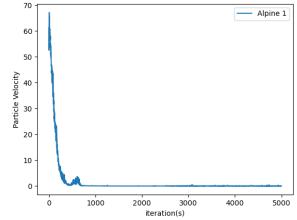
(a)



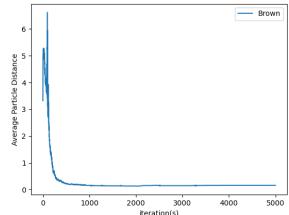
(b)



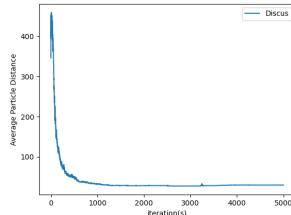
(a)



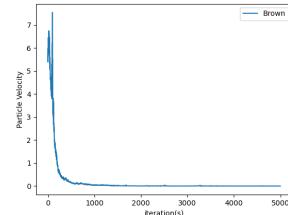
(b)



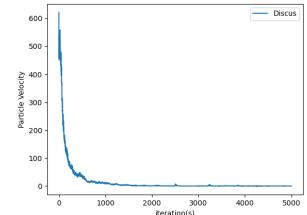
(c)



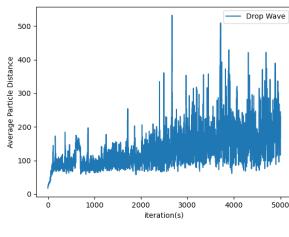
(d)



(c)



(d)



(e)

Fig. 5. (a) Average particle distance for Ackley 1 (b) Average particle distance for Alpine (c) Average particle distance for Brown (d) Average particle distance for Discus (e) Average particle distance for Dropwave

Figure 5 illustrates the average particle distance for the PSO algorithm using Linearly Increasing Grouping across five benchmark functions in a 100-dimensional search space. For Ackley 1, Alpine, Brown, and Discus, swarm diversity drops sharply within the first 500 iterations and remains low for the rest of the optimisation process. This early collapse in diversity indicates that the swarm rapidly converges to a narrow region of the search space, limiting further exploration. The Dropwave function displays a contrasting pattern, with highly unstable and increasing diversity over time. This behaviour suggests inconsistent swarm coordination, possibly due to deceptive landscape features that cause the swarm to oscillate rather than settle. Overall, the results show that Linearly Increasing Grouping does not maintain swarm diversity, and instead promotes early exploitation, with limited exploratory capacity in the later stages of optimisation.

Fig. 6. (a) Average velocity magnitude for Ackley 1 (b) Average velocity magnitude for Alpine (c) Average velocity magnitude for Brown (d) Average velocity magnitude for Discus (e) Average velocity magnitude for Dropwave

Figure 6 presents the average velocity magnitude of the swarm over 5000 iterations for the PSO algorithm incorporating Linearly Increasing Grouping. For the Ackley 1, Alpine, Brown, and Discus functions, velocity magnitudes decrease sharply within the first few hundred iterations and approach near-zero values as the optimisation progresses. This rapid decline indicates an early shift from exploration to exploitation, as particles quickly lose momentum once local best positions are identified. The uniform decay across these functions suggests that the velocity update process is tightly constrained by the gradual dimensional activation in the LIG strategy.

The Dropwave function, however, displays markedly different behaviour. Rather than stabilising, the velocity magnitude remains erratic and fluctuates significantly throughout the search process. This instability reflects difficulties in swarm convergence and suggests that particles continue to accelerate in inconsistent directions due to the deceptive landscape structure. The high variability in particle velocities implies a lack of directional consensus within the swarm, further confirming that LIG does not prevent oscillatory behaviour in challenging optimisation surfaces. Overall, the results suggest that while LIG promotes early convergence in structured landscapes, it may result in unstable dynamics in deceptive or multimodal

environments.

C. Subspace Initialisation

This section provides the results for the standard PSO algorithm with subspace initialisation for the problems listed in section IV. The following figures provided the results for the evaluation metrics discussed in section IV.

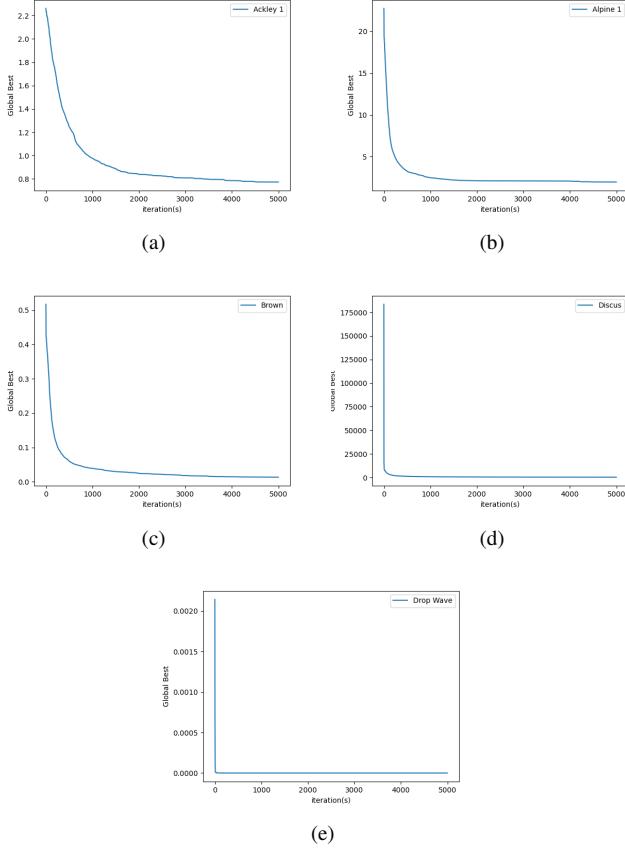


Fig. 7. (a) Global Best for Ackley 1 (b) Global Best for Alpine (c) Global best for Brown (d) Global best for Discus (e) Global best for Dropwave

Figure 7 illustrates the convergence behaviour of the PSO algorithm modified with Subspace Initialisation when applied to five benchmark functions in a 100-dimensional space. For all functions—Ackley 1, Alpine, Brown, Discus, and Dropwave—the global best values improve rapidly within the first few hundred iterations, followed by stable convergence for the remainder of the optimisation. This pattern indicates that Subspace Initialisation enables the swarm to identify promising regions of the search space early in the process. The reduced-dimensional initialisation effectively constrains the search to a lower-dimensional manifold, allowing more focused exploration during the early phase. Once projected into the full-dimensional space, the swarm exploits the identified region with minimal divergence, resulting in smooth and efficient convergence. Notably, even functions with deceptive or ill-conditioned landscapes, such as Discus and Dropwave, converge without exhibiting instability or stagnation. These

results suggest that Subspace Initialisation provides a robust starting configuration that improves convergence consistency and enhances the algorithm's effectiveness on large-scale problems.

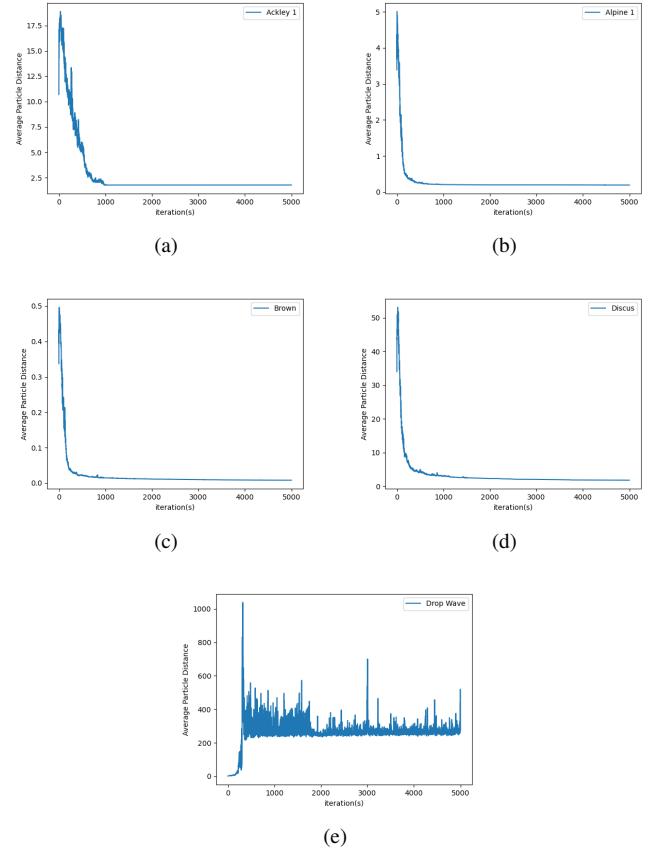


Fig. 8. (a) Average particle distance for Ackley 1 (b) Average particle distance for Alpine (c) Average particle distance for Brown (d) Average particle distance for Discus (e) Average particle distance for Dropwave

Figure 11 presents the average particle distance for the PSO algorithm using Subspace Initialisation, serving as a measure of swarm diversity across five benchmark functions in a 100-dimensional space. For Ackley 1, Alpine, Brown, and Discus, the swarm diversity decreases rapidly during the early iterations and stabilises near zero thereafter. This trend indicates an initial exploratory phase followed by strong convergence behaviour, with minimal divergence in the latter stages. The sharp initial drop in diversity is expected, as Subspace Initialisation confines particles to a low-dimensional region early on, causing them to explore a focused area of the search space before converging. The Dropwave function, however, displays significantly different behaviour. The diversity remains relatively high and fluctuates across the entire run, suggesting that the swarm struggles to settle within a stable region. This behaviour may be attributed to the deceptive nature of the Dropwave function, where narrow minima and steep basins challenge convergence and induce oscillatory swarm dynamics. Overall,

the results show that Subspace Initialisation leads to consistent and structured convergence in well-behaved landscapes, but may still face difficulties in deceptive or highly multimodal problems where more sophisticated diversity control may be needed.

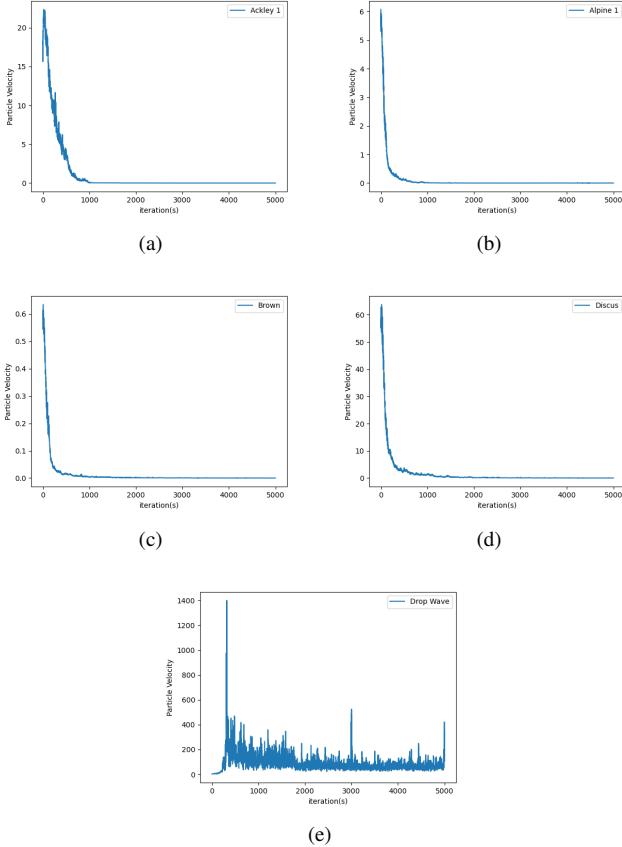


Fig. 9. (a) Average velocity magnitude for Ackley 1 (b) Average velocity magnitude for Alpine (c) Average velocity magnitude for Brown (d) Average velocity magnitude for Discus (e) Average velocity magnitude for Dropwave

Figure 9 illustrates the average velocity magnitude of the swarm throughout the optimisation process for the PSO algorithm with Subspace Initialisation. For the Ackley 1, Alpine, Brown, and Discus functions, particle velocities decrease sharply within the first few hundred iterations and approach near-zero values thereafter. This early drop in velocity indicates a transition from exploration to exploitation as the swarm begins to converge on optimal regions identified during the initial subspace exploration. The stability in velocity following this transition suggests consistent swarm behaviour and minimal directional changes in later iterations, contributing to efficient convergence.

In contrast, the Dropwave function exhibits more erratic velocity behaviour. Although initial velocities also decrease, there are noticeable fluctuations that persist throughout the search. These irregularities reflect unstable swarm dynamics likely caused by the deceptive nature of the Dropwave function, which contains numerous narrow global basins and steep gradients. The observed variability in velocity indicates that

particles continue to accelerate and change direction as they navigate the rugged fitness landscape, making convergence more challenging. Overall, the results confirm that Subspace Initialisation contributes to stable convergence patterns in structured landscapes, while in deceptive environments, additional mechanisms may be necessary to dampen instability and improve directional control.

D. Hybrid

This section provides the results for the hybrid PSO algorithm for the problems listed in section IV. The following figures provided the results for the evaluation metrics discussed in section IV.

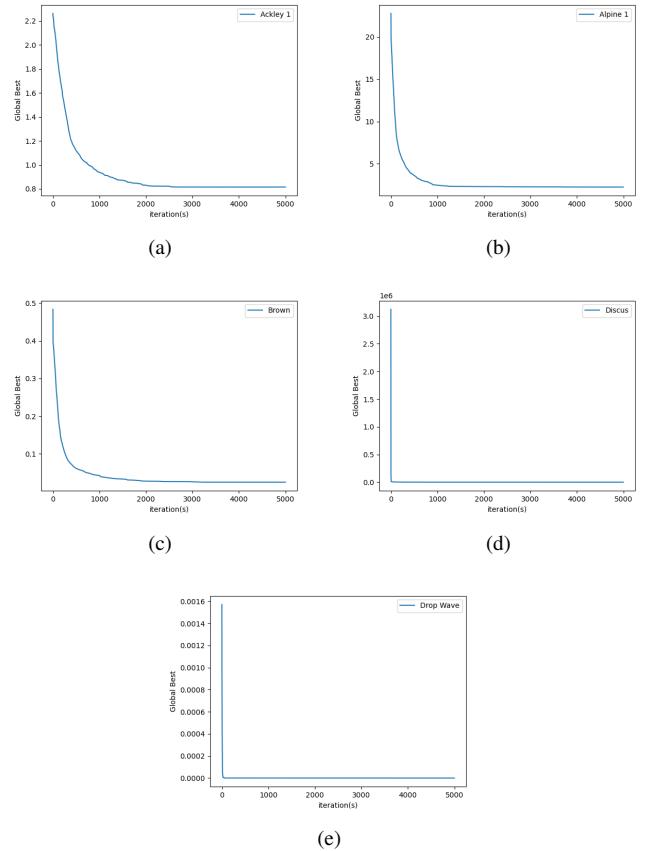


Fig. 10. (a) Global Best for Ackley 1 (b) Global Best for Alpine (c) Global best for Brown (d) Global best for Discus (e) Global best for Dropwave

Figure 10 presents the global best values over 5000 iterations for the hybrid PSO algorithm, which combines Linearly Increasing Grouping and Subspace Initialisation. Across all five benchmark functions—Ackley 1, Alpine, Brown, Discus, and Dropwave—the algorithm exhibits rapid convergence, with global best values stabilising within the first few hundred iterations. This indicates that the hybrid method efficiently exploits early search information, benefiting from the low-dimensional focus of Subspace Initialisation and the gradual dimensional expansion of Linearly Increasing Grouping. For the Discus and Dropwave functions, the convergence behaviour is similar to that observed in the individual strategies.

The swarm rapidly converges, and no further improvements occur beyond early iterations. This suggests that the hybrid approach does not significantly enhance exploration in deceptive or ill-conditioned landscapes. Instead, it maintains the strengths of both techniques in accelerating convergence but may require additional mechanisms to preserve diversity in more complex problem spaces.

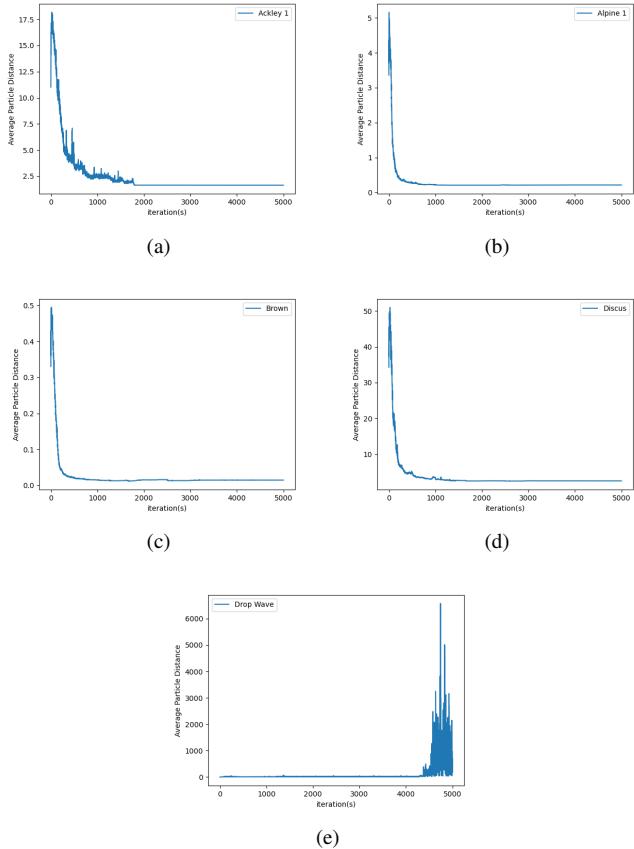


Fig. 11. (a) Average particle distance for Ackley 1 (b) Average particle distance for Alpine (c) Average particle distance for Brown (d) Average particle distance for Discus (e) Average particle distance for Dropwave

Figure 11 displays the average particle distance over time for the hybrid PSO algorithm across five benchmark functions in a 100-dimensional search space. For the Ackley 1, Alpine, Brown, and Discus functions, the swarm diversity decreases sharply within the first few hundred iterations and stabilises near zero for the remainder of the optimisation. This consistent pattern indicates that the hybrid algorithm, like its constituent methods, exhibits rapid convergence and limited exploration in the later stages of the search process. The early collapse in diversity reflects the strong exploitation behaviour introduced by Subspace Initialisation, while the dimensional constraints imposed by Linearly Increasing Grouping contribute to focused, directional movement early in the optimisation.

The Dropwave function shows a different trend, with diversity remaining low initially but increasing erratically in the later iterations. This oscillatory behaviour suggests that the swarm

fails to converge on a stable region of the search space and continues to explore inconsistent directions. This instability is likely a result of the deceptive nature of the Dropwave landscape, which contains narrow and discontinuous optima. Overall, the results indicate that while the hybrid method is effective at rapidly concentrating the swarm in early iterations, it does not significantly enhance exploratory capabilities or prevent instability in deceptive problem domains.

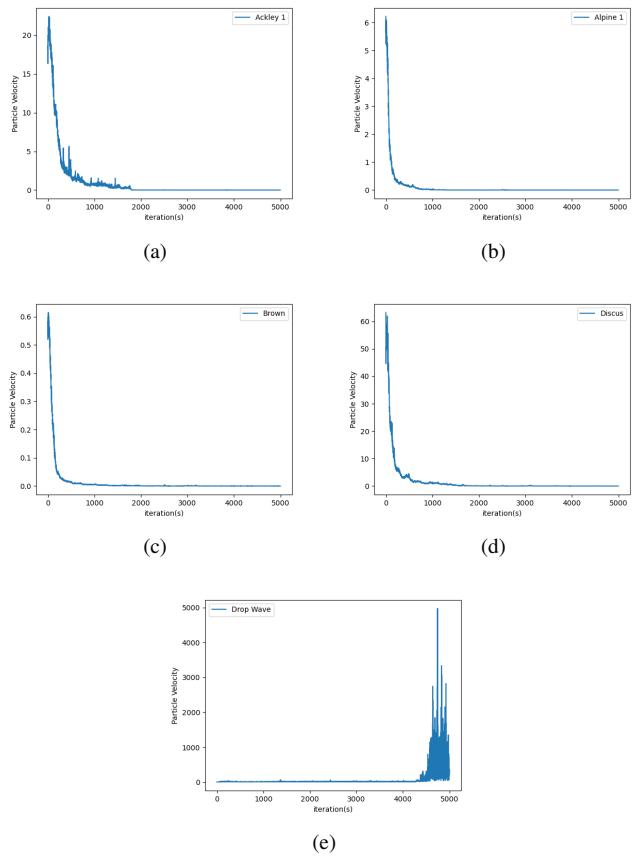


Fig. 12. (a) Average velocity magnitude for Ackley 1 (b) Average velocity magnitude for Alpine (c) Average velocity magnitude for Brown (d) Average velocity magnitude for Discus (e) Average velocity magnitude for Dropwave

Figure 12 illustrates the average velocity magnitude of particles over time for the hybrid PSO algorithm applied to five benchmark functions in a 100-dimensional search space. For Ackley 1, Alpine, Brown, and Discus, velocity magnitudes decrease rapidly within the first few hundred iterations, stabilising near zero in the remainder of the run. This rapid decay indicates a strong exploitation phase triggered early in the optimisation process. The hybrid algorithm's design—combining Subspace Initialisation and Linearly Increasing Grouping—limits movement in early dimensions and concentrates the swarm in a focused region of the search space, which contributes to this early stabilisation in velocity. The Dropwave function exhibits highly erratic and increasing velocity magnitudes across iterations. This behaviour suggests unstable swarm dynamics, likely caused by the deceptive and

rugged nature of the Dropwave landscape. Despite the hybrid method's structured approach, the swarm continues to adjust particle positions aggressively, indicating poor convergence and a lack of directional consensus. These fluctuations reflect an ongoing struggle to identify and exploit the global minimum. Overall, the results show that while the hybrid method promotes rapid velocity stabilisation in structured landscapes, it does not mitigate instability in functions with high modality or sharp discontinuities.

E. Rankings

This section ranks the algorithm based on the performance of the algorithm in the evaluation metrics. To determine if there is a statistical difference between the 3 variations on the standard PSO a Friedman test is used. Consider the table:

TABLE I

TABLE SHOWING THE BEST GLOBAL VALUE FOUND AT THE END OF THE ITERATIONS ON AVERAGE

Algorithm	GB Ackley 1	GB Alpine	GB Brown	GB Discus	GB DropWave	Overall Rank
Standard	4	3	3	3	1	2.8
LIG	1	4	4	4	1	2.8
SubspaceINIT	2	1	1	1	1	1.2
Hybrid	3	2	2	2	1	2

The Friedman test found that for each function there was a statistical difference between the top 3 methods.

VI. CONCLUSION

This paper proposed a hybrid Particle Swarm Optimisation (PSO) algorithm for solving large-scale optimisation problems by integrating Linearly Increasing Grouping and Subspace Initialisation. The results demonstrated that each individual modification improved convergence stability and reduced velocity variance in structured benchmark functions. However, both methods also exhibited early convergence and a loss of swarm diversity in deceptive or multimodal landscapes.

The hybrid method retained the strengths of both approaches, achieving rapid convergence and stable swarm behaviour across most benchmark functions. Nonetheless, it did not significantly improve exploration or prevent oscillatory dynamics in complex functions such as Dropwave. Statistical analysis confirmed that Subspace Initialisation achieved the most consistent performance overall, while the hybrid method provided a competitive balance between convergence speed and general robustness.

Future work may focus on incorporating adaptive diversity preservation strategies or hybridising with local search techniques to enhance performance in deceptive or rugged fitness landscapes.

REFERENCES

- [1] Maurice Clerc. "Particle swarm optimization". In: *International Journal of Numerical Methods* (2002).
- [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] 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.
- [5] 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>.
- [6] E.T. Oldewage, J.P. van Zyl, and A.P. Engelbrecht. "Degrees of Stochasticity in Particle Swarm Optimization". In: *Complex & Intelligent Systems* 5.3 (2019), pp. 239–253. DOI: 10.1007/s40747-019-0103-6.
- [7] 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.
- [8] Yuhui Shi and Russell C. Eberhart. "A modified particle swarm optimizer". In: *Proceedings of the IEEE International Conference on Evolutionary Computation*. IEEE. 1998, pp. 69–73.