PARAMETER
JURNAL MATEMATIKA, STATISTIKA DAN TERAPANNYA

e-ISSN. 2829-6303
Vol. 05 No. 1, April 2026
Pages 45-58

Swarm-Genetics: A Hybrid PSO-GA Regeneration Model for Global Optimization Benchmark Problems

Aprizal Resky[1]*, Zaitun[2], Ryo Hartawan Sasolo[3], Andi Isna Yunita[4]

[1,3]Data Science, Departement of Science, Institut Teknologi Bacharuddin Jusuf Habibie, Parepare, 91122, South Sulawesi, Indonesia
[2]Mathematics, Departement of Science, Institut Teknologi Bacharuddin Jusuf Habibie, Parepare, 91122, South Sulawesi, Indonesia
[4]Departement of Statistics, Faculty of Mathematics and Natural Sciences, Universitas Hasanuddin, Makassar, South Sulawesi, Indonesia

E-mail Correspondence Author: *aprizalresky@ith.ac.id*

Abstract

Particle Swarm Optimization (PSO) is widely used in global optimization due to its simple structure and fast convergence, but it may suffer from premature convergence in multimodal search spaces. This study proposes Swarm-Genetics, an iteration-wise hybrid PSO-GA regeneration framework that combines PSO-based particle movement with Genetic Algorithm operators. In each iteration, particles are first updated using PSO velocity and position equations, then regenerated through crossover and mutation, followed by the selection of the best particles for the next iteration. The proposed method was evaluated on fourteen benchmark functions and compared with standard PSO and GA using mean fitness values. The results show that Swarm-Genetics achieved the lowest mean fitness values across the tested benchmark functions, with several cases producing mean errors close to zero, such as $\mathbf{3.2161 \times 10^{-7}}$, $\mathbf{2.1326 \times 10^{-10}}$, *and* $\mathbf{6.5637 \times 10^{-10}}$. *It also obtained a lower mean value on the Schwefel function than both baseline methods, indicating better exploration in a complex multimodal landscape. These findings provide descriptive numerical evidence that genetic regeneration can improve PSO search performance by enhancing exploration while maintaining exploitation-oriented swarm guidance.*

Keywords: *Genetic algorithm, global optimization, particle swarm optimization, premature convergence, swarm regeneration*

: https://doi.org/10.30598/parameterv5i1pp45-58

Submitted: July 2025
Accepted: April 2026

1. INTRODUCTION

Particle Swarm Optimization (PSO) is one of the optimization techniques inspired by the behaviour of flocks of birds or herds in searching for food sources. In PSO, a group of individuals, referred to as agents or particles, is used to explore the solution space in an effort to find the optimal value of a problem [1]. Although PSO is well known for its simplicity and fast convergence, its search dynamics may become overly concentrated around a limited region of the solution space, especially in multimodal and high-dimensional problems. In such cases, particles tend to lose diversity and repeatedly exploit locally promising regions, which increases the risk of premature convergence. This limitation is particularly relevant in global optimization problems characterized by complex landscapes, numerous local optima, and strong interactions among decision variables. Therefore, a mechanism that not only preserves the fast directional search of PSO but also periodically restores population diversity is highly desirable in order to maintain deeper exploration of the solution space. The importance of balancing convergence speed and diversity has been widely emphasized in PSO theory and in more recent developments of hybrid swarm-based methods [2]. One of the common challenges in applying PSO is when the particles in this algorithm become trapped at specific points in the solution space. This phenomenon is known as the problem of getting stuck at local optima [3]. This problem can hinder the ability of PSO to find the optimal solution or approach a good solution, and it may persist indefinitely at these points, making it difficult to predict when it will escape.

In an effort to address these issues, a lot of research has been conducted to develop this method further, such as adding several optimization stages to the PSO algorithm and combining it with other methods. Many developments in this method have been carried out, such as [4] who developed the inertia weight parameter in the PSO method; [5] who added a reverse predictor to the basic algorithm to avoid premature convergence; [6] who used an ignoring mechanism to avoid premature convergence; [7], who added a multi-stage search strategy to make the search method more effective. [8] modified PSO by adding Cauchy mutation, and [9] applied the second global best to enhance the performance of the original PSO. Research combining PSO with other methods has also been extensively conducted. [3] combined PSO with the Fireworks Algorithm to create a hybrid PS-FW, [10] used MPC-based constraints for hybrid PSO, and [11] combined PSO with the Dipper Throated Optimization algorithm. The combination of PSO with Genetic Algorithm (GA) has also been widely explored. [1] combined PSO with GA by incorporating PSO into GA, similar to [12], who applied PSO into GA for hybrid alternative methods. Additionally, [13] used partitioned populations in PSO and GA for large-scale global optimization problems. The development of PSO methods through combinations or hybrids is extensively used in various problems, especially in conjunction with GA, such as PSO-GA for scheduling problems by [14], structural stretching problems by [15], machining and generator optimization problems [16], [17], [18], [19], and generator dislocation problems in electrical systems by [20].

Genetic algorithms have proven effective in exploring solution spaces more extensively [21], while PSO can provide rapid convergence [22]. This research proposes an innovative and straightforward approach by combining the Particle Swarm Optimization method with genetic algorithms. Our main goal is to harness the strengths and advantages of each method to enhance performance in solving optimization problems. Additionally, we will provide several function tests for the application of the proposed method.

Although hybrid PSO-GA algorithms have been widely studied, many existing approaches treat hybridization mainly as a direct combination of search operators without explicitly defining how swarm diversity is restored after particles become concentrated around the current global best. In multimodal optimization problems, this concentration may reduce the ability of PSO to explore distant regions of the solution space. As a result, the algorithm may repeatedly exploit a locally promising region and fail to escape from local optima. Therefore, the main limitation addressed in this study is not merely the slow convergence of PSO, but the loss of population diversity that occurs during repeated swarm attraction toward the best-known solution.

To address this limitation, this study proposes Swarm-Genetics, an iteration-wise regeneration framework that integrates GA operators into the PSO cycle after particle movement. In contrast to generic PSO-GA hybrid models, the proposed method applies crossover and mutation to the updated particle-position population, while the velocity component of PSO is not directly modified by GA. The regenerated offspring are combined with the current population and then filtered through a selection mechanism so that only the best candidates are retained for the next iteration. Personal best and global best values are updated after this regeneration-selection process. Thus, the novelty of the proposed method lies in the structured interaction between PSO-based directional movement and GA-based population regeneration, which is designed to preserve exploitation while periodically restoring exploratory diversity.

The main contributions of this study are as follows. First, the study formulates an iteration-wise PSO-GA regeneration mechanism for continuous global optimization. Second, the study clarifies the role of GA as a position-level regeneration operator that introduces new search directions without directly changing particle velocities. Third, the proposed framework is evaluated on fourteen benchmark functions with different landscape characteristics to examine its ability to avoid premature convergence and improve solution quality. Fourth, the experimental comparison is reported using repeated independent runs, descriptive statistics, convergence analysis, and statistical testing to support objective performance evaluation.

2. METHOD

To understand and applied the proposed methodology, an algorithm was constructed by two different algorithms. Briefly, in this research explaining the two main algorithms employed in the proposed algorithm as follows.

2.1. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is one of the heuristic optimization methods inspired by the flocking behaviour of birds or fish **[12]**. Each particle moves at a velocity determined by various factors, as naturally present in the particle itself **[3]**. First, there is the influence of particle velocity, referred to as the inertia factor. Second, there is the influence of the particle's memory, considering the previous best position of the particle. Third, there is the influence of the best particle within the group. The following illustrates as **Figure 1** is the movement of particles within the optimization space.

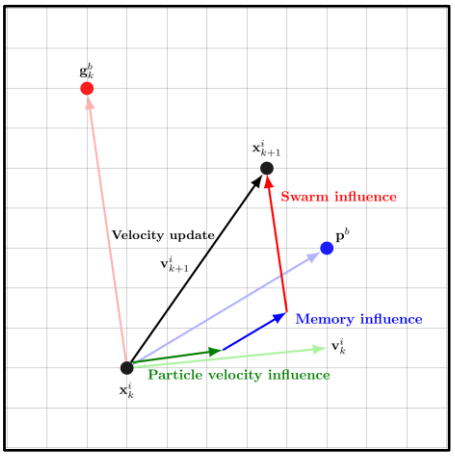

Figure 1. Illustration of particle motion in PSO

Mathematically, the particle movement equation in PSO can be written as follows:

$$\boldsymbol{v}_{k+1}^{i} = w.\boldsymbol{v}_{k}^{i} + c_1 * r_1.\left[\boldsymbol{pb}_{k}^{i} - \boldsymbol{x}_{k}^{i}\right] + c_2.r_2\left[\boldsymbol{gb}_{k} - \boldsymbol{x}_{k}^{i}\right] \quad (1)$$

$$\boldsymbol{x}_{k+1}^{i} = \boldsymbol{x}_{k}^{i} + \boldsymbol{v}_{k+1}^{i} \quad (2)$$

With k is the number iterations and $(i = 1,2,\dots,m)$ are the number of populations for each velocity vector $\boldsymbol{v}_{k}^{i} = \{v_{k}^{i}(1), v_{k}^{i}(2), \dots, v_{k}^{i}(n)\}$ in dimension n and particle position vector $\boldsymbol{x}_{k}^{i} = \{x_{k}^{i}(1), x_{k}^{i}(2), \dots, x_{k}^{i}(n)\}$. During the movement of each particle, every particle has a personal best $\boldsymbol{pb}_{k}^{i} = \{pb_{k}^{i}(1), pb_{k}^{i}(2), \dots, pb_{k}^{i}(n)\}$ and every step of movement every group have a global best $\boldsymbol{gb}_{k} = \{gb_{k}(1), gb_{k}(2), \dots, gb_{k}(n)\}$. Similar to the assumption made by [8],[3] c_1 and c_2 represent cognitive and social influences, respectively; r_1 and r_2 are random numbers in range [0,1], and w is the weight of inertia, which depends on the number of movements.

The mechanism of original PSO [8], [9] process is shown by the **Error! Reference source not found.** below:

Create value of c_1 and c_2 in **Equation (1).**
Generate the initial value and velocity of m particle in population
Calculate fitness function for each particle
Initialized the *Pbest* path and value for each particle by the initial
Taking *Gbest*, both path and value by the best of *Pbest*
  While k < $\boldsymbol{k_{max}}$
    For $\boldsymbol{i = 1, 2 \dots m}$
    Update velocity by **Equation** Error! Reference source not found..
    Update value of particle by **Equation (2).**
    Calculate fitness function
    Update *Pbest* if $Pbest(k+1) < Pbest(k)$
    Update *Gbest* if there $Pbest < Gbest$
Print *Gbest* path
Print *Gbest* value

Algorithm 1. Pseudocode of PSO

An illustration of the motion of each particle in the solution space with a trapped point is shown in the **Figure 2** below.

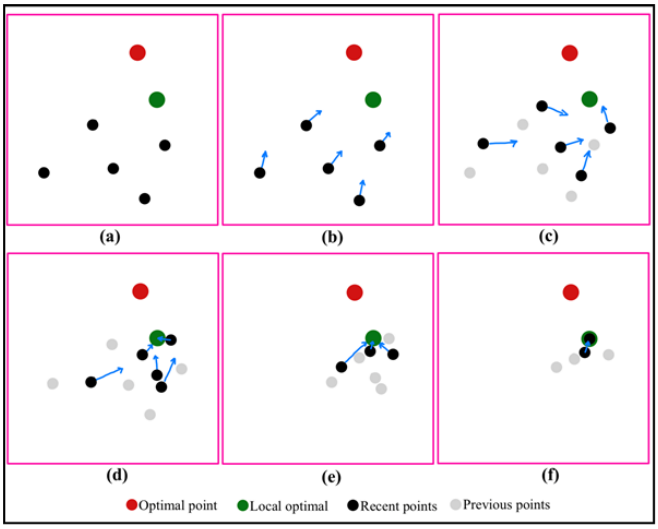

Figure 2. Illustration of populations motion in PSO

2.2. Genetics Algorithm (GA)

The Genetic Algorithm (GA) still draws inspiration from particles undergoing a mechanism of regeneration or evolution [21]. The original mechanism of GA is presented in[1], [12], [21], [23] and is written in the **Algorithm 2** below.

Initialized m chromosome in population as initial value
Calculate fitness
Initialized the probabilities of crossover (pc) and mutation (pm)
Create current population as the first generation
    While k < k_{max}
        crossover: Generate child and calculate the fitness
        mutation: the child is growing up and calculate the fitness
        selection: adding new child in previous generation and select m best
            all Chromosome by sorting the fitness value
        The m chromosome had been creating as new generation
Print solution from the best chromosome
Print fitness value of the best chromosome

Algorithm 2. Pseudocode of GA

To illustrate the regeneration of chromosomes in GA, the **Figure 3**, below is shows it

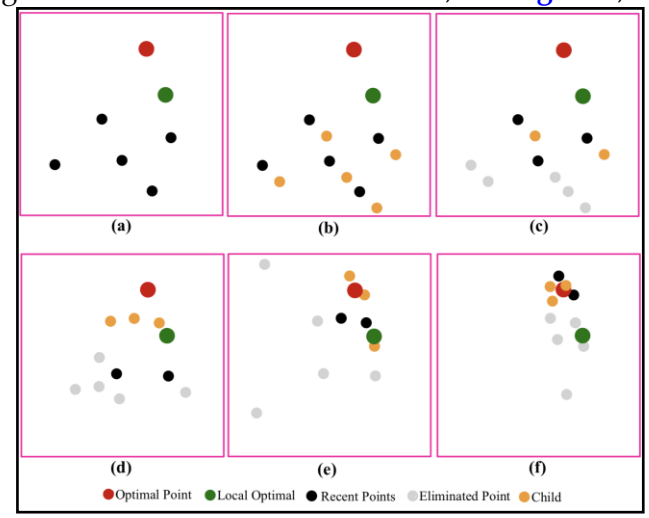

Figure 3. Illustration of populations motion in GA

2.3. Proposed Algorithm

The proposed Swarm-Genetics algorithm is designed as an iteration-wise hybrid framework that combines the movement mechanism of Particle Swarm Optimization (PSO) with the regeneration mechanism of Genetic Algorithm (GA). The main objective of this hybridization is to preserve the fast directional search capability of PSO while increasing population diversity through genetic regeneration.

Below are the step-by-step details of this algorithm:

1. Let $X_i^k = \left(x_{i1}^k, x_{i2}^k, \dots, x_{iD}^k\right)$ denote the position of the i-th particle at iteration k in a D −dimensional search space, and let $V_i^k = \left(v_{i1}^k, v_{i2}^k, \dots, v_{iD}^k\right)$ denote its velocity vector. Each particle stores its personal best position, denoted as $Pbest_i$, while the swarm stores the best solution found by the entire population, denoted as $Gbest$.
2. Evaluate the fitness value $f\left(X_i^0\right)$ for each particle. Set the intial personal best $Pbest_i = X_i^0$, and determine the initial global best $Gbest$ from $Pbest_i$.
3. For each iteration $k = 1,2, \dots, K$, update each particle velocity V_i^k using **Equation** Error! Reference source not found. and **Equation** Error! Reference source not found., followed by the recalculation of the fitness function for the updated particle values. This causes the particles to move or shift from their previous positions
4. Apply GA crossover to selected updated particle positions to generate offspring candidate positions. Apply GA mutation to the offspring candidate posititon to increase population diversity. Boundary violations are repaired by returning infeasible values to the permitted search interval.
5. Evaluate the fitness values of all offspring candidate positions. Combine the updated PSO particle positions and the GA-generated offspring positions into one candidate pool.
6. Select the best N candidate positions based on their fitness values. Update $Pbest_i$ by comparing eact selected particle position with its previous personal best. Update $Gbest$ if any selected particle position has a better fitness value than the previous global best.
7. In the next step, the process described in (2)-(6) is repeated until the maximum number of iterations K is reached or another stopping criterion is satisfied.
8. The best solution $Gbest$ and its fitness value $f(Gbest)$.

To facilitate the observation of the algorithm process, the algorithm flowchart is presented in the following as **Figure 4.**

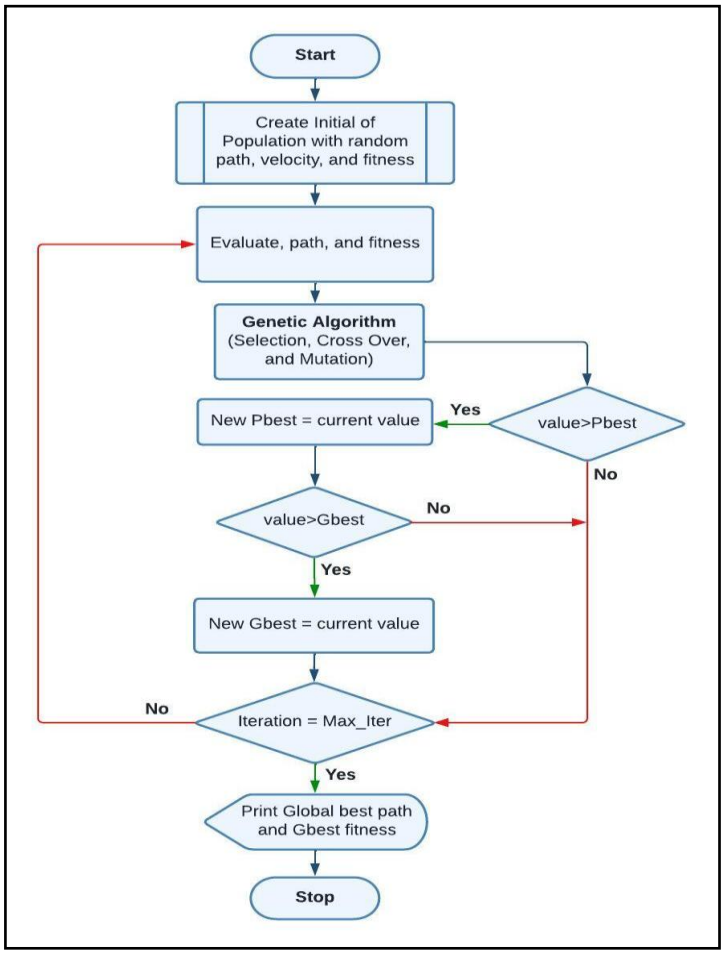

Figure 4. Flowchart of Swarm-Genetics Method

And as an illustration of the particle movement in this algorithm, it can be observed in the following diagram as FIGURE 1.5, which is depicted in the following two-dimensional space.

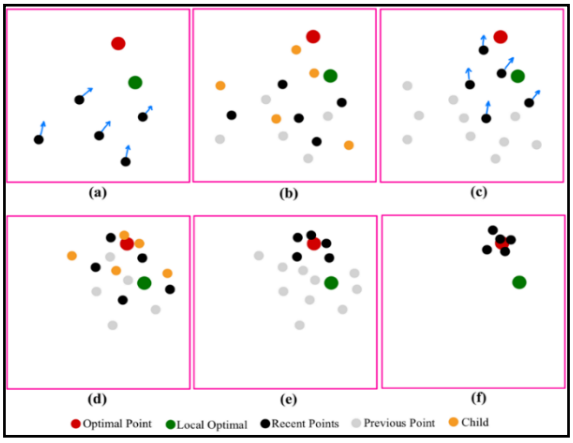

Figure 5. Illustration of population motion in PSO with GA combination

2.3. Benchmark Test Function

The algorithm developed in this research was subsequently applied to 14 different function tests form **[4]**, as indicated in **Table 1**, with the optimal values also displayed in **Table 1** Almost all the function tests used in this study had local optimal points, which could potentially lead to the original algorithm getting trapped in local optima.

Table 1. Test function benchmark

Name	Function	Domain	Global Optimum
Linier	$f_1(\boldsymbol{x}) = \sum_{i=1}^{n} \lvert x_i \rvert$	$[-100,100]^n$	0
Sphere	$f_2(\boldsymbol{x}) = \sum_{i=1}^{n} (x_i)^2$	$[-100,100]^n$	0
Sum square	$f_3(\boldsymbol{x}) = \sum_{i=1}^{n} i(x_i)^2$	$[-10,10]^n$	0
Schumer Steiglitz	$f_4(\boldsymbol{x}) = \sum_{i=1}^{n} (x_i)^4$	$[-100,100]^n$	0
Alpine function	$f_5(\boldsymbol{x}) = \sum_{i=1}^{n} x_i \sin(x_i) + 0.1x_i$	$[-10,10]^n$	0
Schwefel's problem 2.26	$f_6(\boldsymbol{x}) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	$[-500,500]^n$	$-418.98n$
Powell sum	$f_7(\boldsymbol{x}) = \sum_{i=1}^{n} \lvert x_i^{i+1} \rvert$	$[-1,1]^n$	0
Rosenbrock	$f_8(\boldsymbol{x}) = \sum_{i=1}^{n} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	$[-30,30]^n$	0
Levy function	$y_i = 1 + \frac{x_i - 1}{4}$ $f_9(\boldsymbol{x}) = \sin^2(\pi y_1) + \sum_{i=1}^{n} (y_i - 1)^2 (1 + 10 \sin^2(\pi y_i + 1))$	$[-10,10]^n$	0
Rotated Hyper-Ellipsoid	$f_{10}(\boldsymbol{x}) = \sum_{i=1}^{n} \sum_{j=1}^{i} x_j^2$	$[-65.5,65.5]^n$	0
Rastrigin	$f_{11}(\boldsymbol{x}) = 10n + \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i)]$	$[-5.12,5.12]^n$	0
Non-Continuous Rastrigin	$f_{12}(\boldsymbol{x}) = \sum_{i=1}^{n} y_i^2 - 10\cos(2\pi y_i) + 10$ $y_i = \begin{cases} x_i & x_i < 0.5 \\ \frac{round(2x_i)}{2} & \lvert x_i \rvert \geq 0.5 \end{cases}$	$[-5.12,5.12]^n$	0
Zakharov	$f_{13}(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^4$	$[-5,10]^n$	0
Weierstrass	$f_{14}(\boldsymbol{x}) = \sum_{i=1}^{n} \left[\sum_{k=0}^{kmax} \left(a^k \cos\left(2\pi b^k (x_i + 0.5)\right)\right) - n \sum_{k=0}^{kmax} a^k \cos(\pi b^k) \right]$	$[-0.5,0.5]^n$	0

The application of the combined PSO and GA algorithm adopts the framework of the original algorithms and incorporates several additional parameters for consideration in **Table 2** To ensure a fair comparison, the parameters used in this algorithm are set to be the same.

Table 2. Parameter setting in PSO

Process	Parameter
PSO[3]	$N = 30, n = 20, w(t) = w_{max} - t\frac{w_{max} - w_{min}}{I_{max}}, w_{max} = 0.95,$ $w_{min} = 0.4, c_1 = c_2 = 1.45, I_{max} = 200$
GA	$N = 30, n = 20, pc = 1, pm = 0.4, I_{max} = 200$
Swarm-Genetics	$N = 30, n = 20, w(t) = w_{max} - t\frac{w_{max} - w_{min}}{I_{max}}, w_{max} = 0.95,$ $w_{min} = 0.4, c_1 = c_2 = 1.45, \quad pc = 1, pm = 0.4, I_{max} = 200$

In the genetic algorithm process, this research employs the following scenarios for the crossover and mutation stages:

- **Cross-Over**: Given the best m chromosomes sorted by their fitness values, assume the population is represented $\boldsymbol{x}^i = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^m\}$, Then, a cross-over operation is performed as follows: ($\boldsymbol{x}^1$ and $\boldsymbol{x}^2$), ($\boldsymbol{x}^3$ and $\boldsymbol{x}^4$), … , and ($\boldsymbol{x}^{m-1}$ and $\boldsymbol{x}^m$). Consider $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$ as parent to perform cross over resulting $\boldsymbol{y}^1 = \big(y^1(1), y^1(2), \dots, y^1(n)\big)$ and $\boldsymbol{y}^2 = \big(y^2(1), y^2(2), \dots, y^2(n)\big)$, by selecting a random number $co = random([2, n-1])$ and then swapping genes between $\boldsymbol{x}^1 = \big(x^1(1), x^1(2), \dots, x^1(n)\big)$ and $\boldsymbol{x}^2 = \big(x^2(1), x^2(2), \dots, x^2(n)\big)$ the result of the cross-over is $\boldsymbol{y}^1 = \big(x^2(1), x^2(2), \dots, x^2(co), x^1(co+1), x^1(co+2), x^1(n)\big)$ and $\boldsymbol{y}^2 = \big(x^1(1), x^1(2), \dots, x^1(co), x^2(co+1), x^2(co+2), x^2(n)\big)$. This process generates m new chromosomes $\boldsymbol{y}^i = \{\boldsymbol{y}^1, \boldsymbol{y}^2, \dots, \boldsymbol{y}^m\}$ which will mutated.
- **Mutation**: Given the m chromosomes resulting from the cross-over $\boldsymbol{y}^i = \{\boldsymbol{y}^1, \boldsymbol{y}^2, \dots, \boldsymbol{y}^m\}$, these chromosomes undergo mutation through two processes. First, a mutation similar to **[21]**, consider chromosome $\boldsymbol{y}^1$ the mutation results is $\boldsymbol{z}^1$ selected as the best fitness value from the following three chromosomes.

$$\{\boldsymbol{y}^1, \boldsymbol{y}^1 + \boldsymbol{\varepsilon}, \boldsymbol{y}^1 - \boldsymbol{\varepsilon}\}$$

where $\boldsymbol{\varepsilon} = \big(\varepsilon(1), \varepsilon(2), \dots, \varepsilon(n)\big)$ is randomly chosen within the interval $\left[-\frac{1}{2k}, \frac{1}{2k}\right]$ with k being the current iteration or generation value. This mutation process yields m mutated chromosomes, $\boldsymbol{z}^i = \{\boldsymbol{z}^1, \boldsymbol{z}^2, \dots, \boldsymbol{z}^m\}$, which are updated. Then, the second process is carried out by selecting $mo = random\big([1, 1 + fix\big(n * (1 - pm)\big)]\big)$, the value of $z(mo), z(mo+1), \dots, z(mo+n-1)$ will be update with the following equation

$$[z(mo), z(mo+1), \dots, z(mo+n-1)] := cl * [z(mo), z(mo+1), \dots, z(mo+n-1)]$$

where cl is the parameter convergence acceleration which satisfied with equation

$$cl = \frac{1}{(k+1)^2}.$$

This second process is applied to each chromosome by generating mo randomly, resulting a total of $2m$ chromosomes. The final step of the mutation process involves selecting the best m mutated chromosomes from the $2m$ chromosomes, resulting as $\boldsymbol{w}^i = \{\boldsymbol{w}^1, \boldsymbol{w}^2, \dots, \boldsymbol{w}^m\}$.

3. RESULTS AND DISCUSSION

The numerical results for each test function obtained using the combination of PSO and GA algorithms are presented sequentially in accordance with the test functions listed in **Table 1** The performance of the proposed algorithm in this research effectively mitigates the instances where the PSO algorithm becomes trapped in local optima.

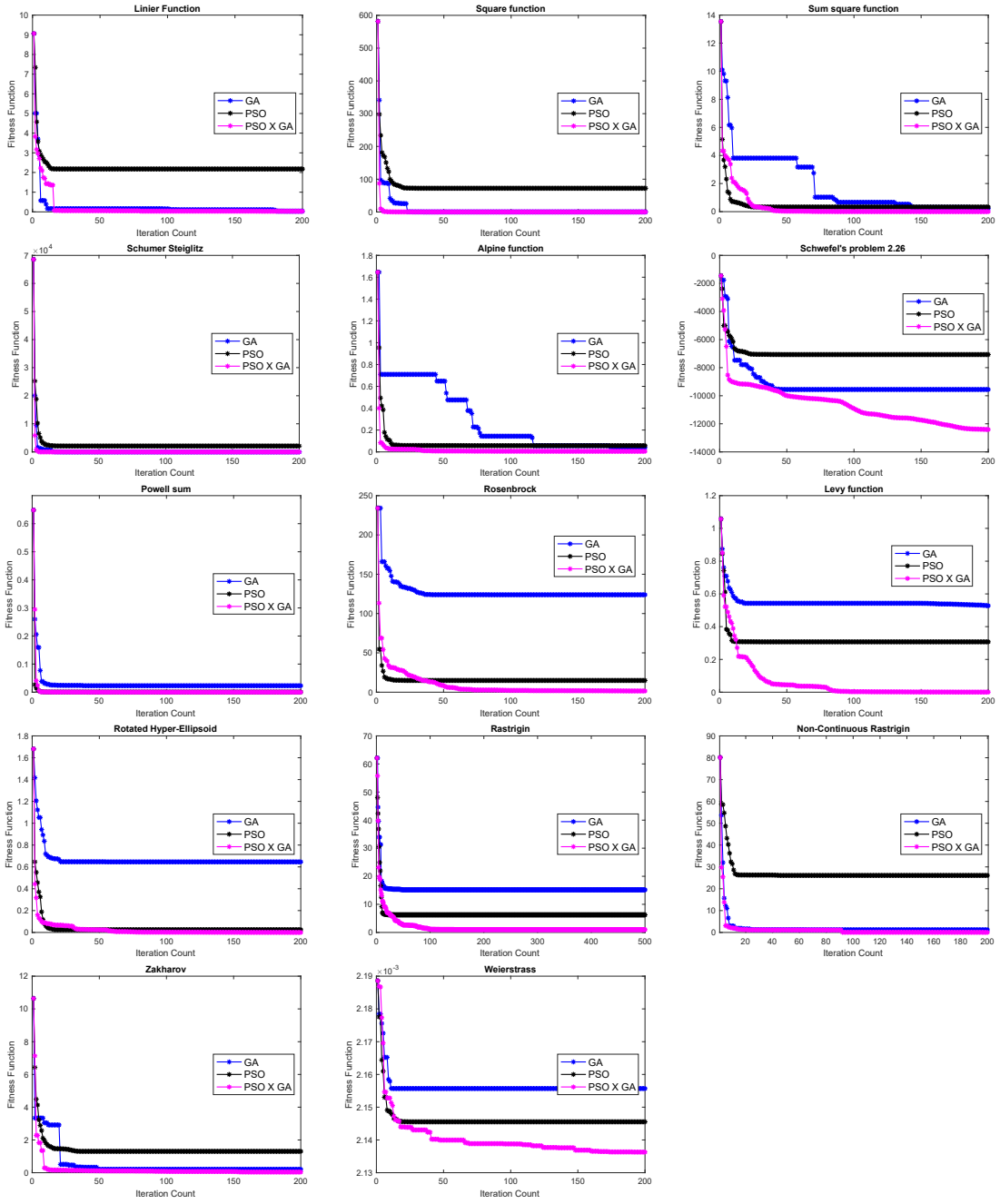

Figure 6. Graph of performance GA, PSO, and Swarm-Genetics for all function

Table 3. Mean of the result of PSO, GA, and Swarm-Genetics for each function

Test function	PSO $(\bar{x})$	GA $(\bar{x})$	Swarm-Genetics $(\bar{x})$
f_1	$1.8976E+00.$	$8.2656E-04.$	$3.2161E-07.$
f_2	$5.2271E+01.$	$1.1219E-03.$	$2.1326E-10.$
f_3	$1.2345E-01.$	$9.8116E-02.$	$7.3225E-07.$
f_4	$1.5244E+01.$	$8.5621E-06.$	$1.7892E-07.$
f_5	$1.2113E-01.$	$7.3489E-01.$	$6.5637E-10.$
f_6	$-7.1252E+03.$	$-9.1178E+03.$	$-1.3754E+04.$
f_7	$3.9981E-02.$	$8.8264E-02.$	$8.7811E-04.$
f_8	$1.4716E+01.$	$8.9425E+01.$	$3.1152E-02.$
f_9	$2.4153E-01.$	$4.6721E-01.$	$7.1673E-09.$
f_{10}	$3.8172E-02.$	$6.7118E-01.$	$6.8876E-05.$
f_{11}	$7.3721E+00.$	$1.2776E+02.$	$1.6254E-01.$
f_{12}	$2.4991E+01.$	$7.1266E-01.$	$2.5482E-08.$
f_{13}	$7.7281E+01.$	$2.5511E-01.$	$3.1667E-04.$
f_{14}	$2.1499E-03.$	$2.1577E-03.$	$2.1131E-03.$

According to the results of the computational tests in **Figure 6** for each function, PSO (Particle Swarm Optimization) often gets trapped in local optima several times.Therefore, in this research, a proposed algorithm proves to be quite effective and efficient in locating global optima while effectively avoiding computational processes trapped in local optima. And by the performance results in **Table 3**, with the average fitness values for each algorithm in global optimization problems with a dimension of $n = 20$. Several factors contribute to the effectiveness of this algorithm, which leverages the strengths of both PSO and GA while addressing their respective weaknesses. PSO excels in particle behaviour by focusing on following the best-performing particle within the population. However, its drawback is that this best-performing particle does not guarantee proximity to the global optimum. This hybrid approach proves to be highly effective, efficient, and sufficiently accurate for addressing global optimization problems **[1], [2], [2], [3].**

4. CONCLUSION

This study proposed Swarm-Genetics, an iteration-wise hybrid PSO-GA regeneration framework for continuous global optimization. The method combines PSO-based directional movement with GA-based regeneration of particle positions through crossover, mutation, and selection. In the proposed framework, GA operates on the updated particle-position population after the PSO movement stage, while particle velocities are not directly modified by GA. This design aims to maintain the exploitation capability of PSO while restoring population diversity to reduce premature convergence. Experimental evaluation on fourteen benchmark functions showed that the proposed method achieved the lowest mean fitness values across all tested functions, with several benchmark functions producing near-zero mean errors, such as 3.2161×10^{-7}, 2.1326×10^{-10}, and 6.5637×10^{-10}. The results indicate that Swarm-Genetics is particularly beneficial for multimodal benchmark functions where local optima can trap standard PSO. Future work may extend the framework by using adaptive regeneration

probability, diversity-based triggering mechanisms, and validation on real-world engineering optimization problems.

Acknowledgments

The author would like to express sincere appreciation to Institut Teknologi Bacharuddin Jusuf Habibie for the institutional support and facilitation provided during the completion of this research.

Funding Information

This research was funded by Institut Teknologi Bacharuddin Jusuf Habibie under the research grant scheme with contract number 005/IT13.D/KH-LPPM/PDP/2025.

Author Contributions Statement

Aprizal Resky served as the corresponding author and was responsible for all correspondence related to the manuscript. He contributed significantly to the conceptualization, methodology, validation, formal analysis, investigation, resource management, original draft writing, review and editing, supervision as well as project administration and funding acquisition. Zaitun participated in the conceptualization and methodology, supported software development and original draft writing, and assisted in the review and editing process. Ryo Hartawan S. contributed to the validation, original draft writing and was involved in the review and editing process, visualization, and project administration. Andi Isna Yunita was responsible for methodology, validation, formal analysis, review and editing, and visualization.

Conflict of Interest Statement

The authors declare that there are no known conflicts of interest that could have influenced the findings presented in this study.

Data Availability

Data availability is not applicable to this paper, as no new data were created or analyzed in this study.

REFERENCES

[1] J. Sharma and R. S. Singhal, " Comparative research on genetic algorithm, particle swarm optimization and hybrid GA-PSO ," *2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 110–114, 2015, Accessed: Jul. 26, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/7100231/authors#authors

[2] Y. Liu *et al.*, "Optimization of five-parameter BRDF model based on hybrid GA-PSO algorithm," *Optik (Stuttg)*, vol. 219, p. 164978, Oct. 2020, doi: 10.1016/j.ijleo.2020.164978.

[3] S. Chen, Y. Liu, L. Wei, and B. Guan, "PS-FW: A Hybrid Algorithm Based on Particle Swarm and Fireworks for Global Optimization," *Comput Intell Neurosci*, vol. 2018, 2018, doi: 10.1155/2018/6094685.

[4] A. Nickabadi, M. M. Ebadzadeh, and R. Safabakhsh, "A novel particle swarm optimization algorithm with adaptive inertia weight," *Appl Soft Comput*, vol. 11, no. 4, pp. 3658–3670, Jun. 2011, doi: 10.1016/j.asoc.2011.01.037.

[5] L. Li, F. Liu, G. Long, P. Guo, and X. Bie, "Modified particle swarm optimization for BMDS interceptor resource planning," *Applied Intelligence*, vol. 44, no. 3, pp. 471–488, Apr. 2016, doi: 10.1007/s10489-015-0711-9.

[6] C.-F. Wang and K. Liu, "A Novel Particle Swarm Optimization Algorithm for Global Optimization," *Comput Intell Neurosci*, vol. 2016, pp. 1–9, 2016, doi: 10.1155/2016/9482073.

[7] B. Han, B. Li, and C. Qin, "A novel hybrid particle swarm optimization with marine predators," *Swarm Evol Comput*, vol. 83, p. 101375, Dec. 2023, doi: 10.1016/j.swevo.2023.101375.

[8] H. Wang, C. Li, Y. Liu, and S. Zeng, "A Hybrid Particle Swarm Algorithm with Cauchy Mutation," in *2007 IEEE Swarm Intelligence Symposium*, IEEE, Apr. 2007, pp. 356–360. doi: 10.1109/SIS.2007.367959.

[9] Y.-B. Shin and E. Kita, "Search performance improvement of Particle Swarm Optimization by second best particle information," *Appl Math Comput*, vol. 246, pp. 346–354, Nov. 2014, doi: 10.1016/j.amc.2014.08.013.

[10] P. A. Gbadega and Y. Sun, "A hybrid constrained Particle Swarm Optimization-Model Predictive Control (CPSO-MPC) algorithm for storage energy management optimization problem in micro-grid," *Energy Reports*, vol. 8, pp. 692–708, Nov. 2022, doi: 10.1016/j.egyr.2022.10.035.

[11] M. Y. Shams, E.-S. M. El-kenawy, A. Ibrahim, and A. M. Elshewey, "A hybrid dipper throated optimization algorithm and particle swarm optimization (DTPSO) model for hepatocellular carcinoma (HCC) prediction," *Biomed Signal Process Control*, vol. 85, p. 104908, Aug. 2023, doi: 10.1016/j.bspc.2023.104908.

[12] T. Mahardhika, "Hybrid Algorithm as alternative method for optimization, a combination Genetic Algorithm and Particle Swarm Optimization," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Feb. 2021. doi: 10.1088/1742-6596/1764/1/012040.

[13] A. F. Ali and M. A. Tawhid, "A hybrid particle swarm optimization and genetic algorithm with population partitioning for large scale optimization problems," *Ain Shams Engineering Journal*, vol. 8, no. 2, pp. 191–206, Jun. 2017, doi: 10.1016/j.asej.2016.07.008.

[14] A. Amirteimoori, I. Mahdavi, M. Solimanpur, S. S. Ali, and E. B. Tirkolaee, "A parallel hybrid PSO-GA algorithm for the flexible flow-shop scheduling with transportation," *Comput Ind Eng*, vol. 173, p. 108672, Nov. 2022, doi: 10.1016/j.cie.2022.108672.

[15] J. Li *et al.*, "Predicting the shear strength of concrete beam through ANFIS-GA–PSO hybrid modeling," *Advances in Engineering Software*, vol. 181, p. 103475, Jul. 2023, doi: 10.1016/j.advengsoft.2023.103475.

[16] Q. Zhang, R. M. Ogren, and S.-C. Kong, "A comparative study of biodiesel engine performance optimization using enhanced hybrid PSO–GA and basic GA," *Appl Energy*, vol. 165, pp. 676–684, Mar. 2016, doi: 10.1016/j.apenergy.2015.12.044.

[17] C. Li, R. Zhai, H. Liu, Y. Yang, and H. Wu, "Optimization of a heliostat field layout using hybrid PSO-GA algorithm," *Appl Therm Eng*, vol. 128, pp. 33–41, Jan. 2018, doi: 10.1016/j.applthermaleng.2017.08.164.

[18] L. Shi, J. Gong, and C. Zhai, "Application of a hybrid PSO-GA optimization algorithm in determining pyrolysis kinetics of biomass," *Fuel*, vol. 323, p. 124344, Sep. 2022, doi: 10.1016/j.fuel.2022.124344.

[19] N. Chityal and S. Sapkal, "Performance analysis of GA, PSO and JA for determining the optimal parameters in friction drilling process," *Engineering Science and Technology, an International Journal*, vol. 35, Nov. 2022, doi: 10.1016/j.jestch.2022.101246.

[20] M. H. Moradi and M. Abedini, "A combination of genetic algorithm and particle swarm optimization for optimal DG location and sizing in distribution systems," *International Journal of Electrical Power and Energy Systems*, vol. 34, no. 1, pp. 66–74, Jan. 2012, doi: 10.1016/j.ijepes.2011.08.023.

[21] M. L. Shahab, F. Azizi, B. A. Sanjoyo, M. I. Irawan, N. Hidayat, and A. M. Rukmi, "A genetic algorithm for solving large scale global optimization problems," in *Journal of*

Physics: Conference Series, IOP Publishing Ltd, Mar. 2021. doi: 10.1088/1742-6596/1821/1/012055.

[22] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Inf Process Lett*, vol. 85, no. 6, pp. 317–325, Mar. 2003, doi: 10.1016/S0020-0190(02)00447-7.

[23] X. Luo, Q. Qian, and Y. F. Fu, "Improved genetic algorithm for solving flexible job shop scheduling problem," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 480–485. doi: 10.1016/j.procs.2020.02.061.