Biology and Medical Engineering
ISSN: 3080-7204| E-ISSN: 3080-7212
Volume 1 , Issue 3

Article

Analysis of PSO Algorithm and GA

Chen Lin*, Mengmeng Shen*

Huaibei Institute of Technology, School of Electronics and Information Engineering, Huaibei, 235000, China.

***Corresponding author:** Chen Lin, 2912479663@qq.com; Mengmeng Shen, 2761417993@qq.com.

CITATION

Lin C and Shen MM. Analysis of PSO Algorithm and GA. Biology and Medical Engineering. 2025; 1(3): 146.

https://doi.org/10.63808/bme.v1i3.146

ARTICLE INFO

Received: 25 June 2025
Accepted: 13 August 2025
Available online: 15 November 2025

Abstract: Particle Swarm Optimization (PSO) and Genetic algorithm (GA) are widely used in real life and are constantly improved and integrated. Both of them have their own advantages in different fields. Both of them have their own advantages in different fields. It performs exceptionally well in solving complex nonlinear problems. This article will conduct a systematic analysis from the perspectives of algorithm principles, performance comparison, and application scenarios, and also explore the future development trends by combining real-world cases.

Keywords: optimization algorithm principle; GA principle; performance comparison; application scenarios

1. Introduction

With the continuous development of modern society and the continuous advancement of technology, a variety of complex problems have emerged, which has led to the introduction of traditional algorithms. In numerous fields such as engineering projects, economic data management, and computer communication, it is often necessary to complete various tasks with greater accuracy and efficiency. The emergence of the PSO algorithm and the GA, as two important intelligent optimization algorithms, has highlighted their outstanding performance in solving these problems. For instance, it improved circuit performance and optimally planned logistics distribution routes, cutting costs and shortening delivery time. In industry, proper production process arrangement and resource allocation enhanced production efficiency.

However, both algorithms have their own advantages and disadvantages and also have certain limitations. The PSO algorithm is prone to getting stuck in local optima and cannot evenly cover the entire space. Meanwhile, the convergence speed of the GA is relatively slow, and the fitness value becomes relatively lower as the number of iterations increases. This article will analyze the principles of PSO algorithm and GA, compare their performance, briefly introduce their application scenarios, and provide more effective methods and ideas for problem optimization in the future.

2. Analysis of the Principles of PSO Algorithm and GA

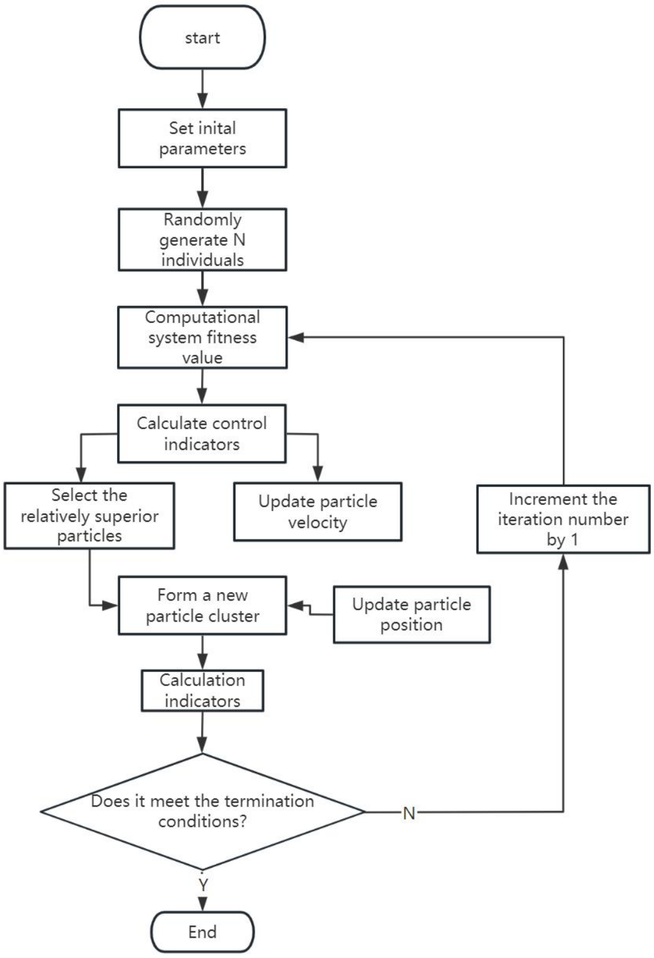
2.1. Principle of Optimized Algorithm

The PSO algorithm, proposed by Kennedy and Eberhart in 1995, assumes each particle as a potential optimal solution in the dimensional space. Particles update positions based on current locations to reach the optimum. They track individual best positions as individual optima, and the group records the overall best as the global optimum (Huber et al., 2024; Hu et al., 2024). The PSO algorithm converges relatively fast and performs well in continuous optimization. However, it tends to get trapped in local optima, struggling with complex multi - peaked functions, thus requiring improvement.

2.1.1 Flowchart of PSO Algorithm

Figure 1

Flowchart of PSO Algorithm



As shown in the PSO algorithm flowchart in **Figure 1**, first set initial parameters and randomly generate N individuals. Calculate system fitness, then compute control indicators to select better particles for position updating, forming a new swarm. Meanwhile, update particle velocities (Gupta and Chak, 2024; Saini et al., 2023)

2.1.2 Standard Particle Swarm Expression

In another space, there are N populations. Then the position of the i-th particle is $x_i=(x_{i1},x_{i2},\ldots, x_{iD})$, and its velocity is $x_i=(x_{i1},x_{i2},\ldots, x_{iD})$. The iterative velocity of the particle is as follows:

$$v_i^{(t+1)}=w\cdot v_i^t+c_1r_1\times(P^t_{best,i}-x_i^t+c_2\cdot r_2\cdot(G^t_{best}-x_i^t) \quad (1)$$

The inertia weights are as follows:

$$w=w_{max}-\frac{t}{t_{max}}(w_{max}-w_{min}) \quad (2)$$

The updated positions of the particles are as follows:

$$x_i^{t+1}=x_i^t+v_i^{t+1} \tag{3}$$

When $f(x_i)<f(p_i)$, $p_i=x_i$; when $f(x_i)<f(p_g)$, $p_g=x_i$. Due to different conditions, the particles update their optimal positions differently.

As shown in **Table 1** parameter settings, among which, vi is the velocity of particle i at time t, xi is the position of particle i at time t, w is the inertia weight, c_1,c_2 are acceleration constants, usually $c_1=c_2=2$, which controls the optimal positions of individual particles and the entire population.r_1 and r_2 are random numbers within the range of [0, 1].The inertia weight w determines the degree to which particles rely on their own historical velocities. A larger w is conducive to global search, while a smaller w is beneficial for local search. The acceleration constants c_1 and c_2 respectively control the step lengths of the particles' flight towards the individual optimal position and the global optimal position (Abibou et al., 2024; Said and Jawale, 2025).

Table 1

Parameter Settings

Key parameters	v_i	x_i	p_i	p_g	w
Concept	The direction and speed of particle movement	The coordinates of particles in the solution space	The optimal position found by the particle itself	The optimal position found by the particle swarm	Inertia weight, balancing global and local search capabilities

2.1.3 Improvement of PSO Algorithm

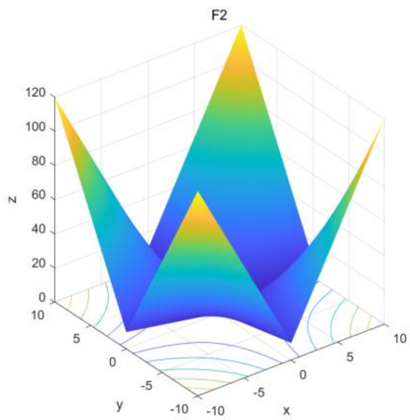
The particle swarm optimization algorithm is widely used, but it has obvious drawbacks: Firstly, it is prone to getting stuck in local optima, with the particle information converging and diversity decreasing in the later stage, making it difficult to escape; Secondly, the convergence accuracy is insufficient, as the heuristic search makes the convergence slow and the accuracy limited when approaching the optimal solution; Thirdly, the parameter selection is difficult, as the parameters have a significant impact on performance and require repeated trials to determine, increasing the difficulty and cost of use (Jin et al., 2021; Jain et al., 2025; Gao, 2025).To avoid local optima, a mutation operation can be introduced to randomly change the

WISDOM ACADEMIC ISSN: 3080-7204| E-ISSN: 3080-7212 Volume 1 , Issue 3

positions of the particles. To accelerate the convergence speed, an adaptive parameter adjustment strategy can be adopted, such as dynamically changing the inertia weight. As shown in **Figure 2**, the objective function of the algorithm optimization is F2. The surface of this function presents a multi-peak shape (multiple elevated and depressed "mountains" and "valleys"), and such functions are often used to test the global optimization ability of the algorithm (it requires escaping from the local optimum and finding the global minimum) (Deng and Liu, 2025; Seghier et al., 2020).

Figure 2

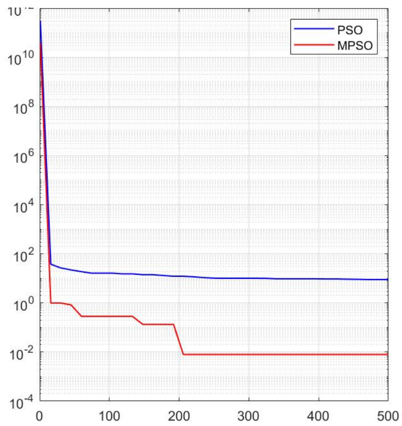
The objective function of the algorithm optimization



As shown in the comparison curves of POS and MPOS in **Figure 3**, PSO (the blue line) has a rapid decrease in fitness values in the early stage, but after approximately 100 iterations, it enters a plateau period, and the fitness values no longer significantly decrease, indicating that the algorithm may have fallen into a local optimum and is unable to find a better solution. While MPSO (the red line) is the improved PSO algorithm, its fitness value keeps decreasing and stabilizes at a much lower level in the later stage (far superior to PSO). This indicates that MPSO enhances its global exploration capability through strategy improvements (such as mutation, weight adjustment, etc.), enabling it to escape from local optima and find solutions that are closer to the global optimum (Laishram et al., 2018; Shui et al., 2025).

Figure 3

Comparison curves of POS and MPOS



2.2. Principle of GA

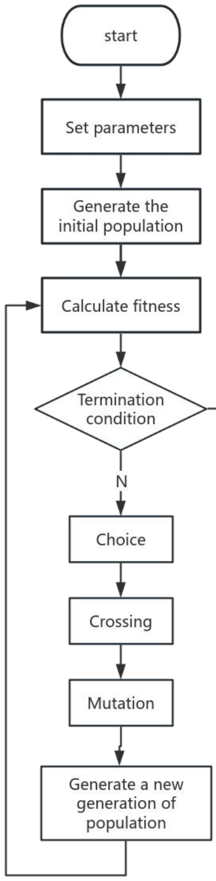
The GA was proposed by John Holland in the 1970s. It mimics the biological evolution mechanism, such as natural selection and gene inheritance, and through iterative evolution, it seeks the optimal solution. It represents the solution to the problem as chromosomes, and multiple chromosomes form a population. The algorithm achieves the evolution of the population through three basic operations: selection, crossover and mutation.GA have strong capabilities for searching globally and can find better solutions in complex solution spaces. However, the computational load of GA is relatively large, their convergence speed is relatively slow, and they are also quite sensitive to the selection of parameters (Gao, 2025).

2.2.1 Flowchart of GA

As shown in **Figure 4** (the GA flowchart), in the initialization stage, start with “Start”, then “Set parameters”, and “Generate initial population”. In the evaluation stage, initiate “fitness calculation” to score each particle based on the objective function. Then, check the “termination condition”. If the max iterations or fitness convergence is met, “output the result” and end; otherwise, enter the genetic operation stage, generate a new population, recalculate fitness, and re - check the termination condition (Kheirdast et al., 2024).

Figure 4

Flowchart of GA



2.2.2 GA part expression

The fitness function F(x) is used to evaluate the quality of individuals (particles), and it is defined according to the specific optimization problem. For example, for the problem of minimizing the objective function f(x), the fitness function can be set as:

$$F(x)=f(x)+\epsilon_i \tag{4}$$

Here, ϵ is a very small positive number, which is used to prevent the denominator from being zero.

The roulette selection method commonly used in the operation is adopted. The probability P_i of individual i being selected:

$$P_i=\frac{F(x_i)}{\sum_{j=1}^{N} F(x_j)} \tag{5}$$

Here, N represents the population size, and F(xi) is the fitness value of individual i.

For the crossover operation, let's take single-point crossover as an example. Randomly select a crossover point k. For the two parent individuals and the offspring individual generated after crossover:

$$x_{c1}=\left[x_{p1}(1:k),x_{p2}(k+1:n)\right] \tag{6}$$

$$x_{c2}=[x_{p2}(1:k),x_{p1}(k+1:n)] \tag{7}$$

Here, x(a:b) represents the part of the vector x from the a-th element to the b-th element. The mutation operation has a mutation probability of Pm. For an n-dimensional vector x, each dimension undergoes mutation with a probability of Pm. For example, in real number encoding, the mutated element can be expressed as $x_i^{'}=x_i+\Delta$, where Δ is a random perturbation.

3. Comparison of Some Performance Characteristics of PSO Algorithm and GA

3.1. Comparison of Convergence Speeds

As shown in the convergence speed comparison curve of the PSO algorithm and the GA in **Figure 5**, the PSO algorithm shows a rapid decrease in the objective function value in the early iterations, quickly approaching 0 and remaining stable thereafter. The GA algorithm also shows a downward trend, but its initial decline rate is slower than that of PSO. Moreover, it only converges gradually after continuous iterations within a higher range of the objective function value. In terms of convergence speed, the PSO algorithm converges significantly faster and stabilizes after approximately 20 iterations. The GA convergence process is relatively slow and requires more iterations to approach the optimal solution, demonstrating the advantage of PSO in rapid optimization. The function selected for the experimental test is as follows:

Sphere function (unimodal function, used to test the basic convergence performance and reflect the optimization efficiency of simple problems):

$$f(x)=\sum i=1nx_{i2} \tag{8}$$

Rastrigin function (multi-modal function, used to test global search ability and avoid local optimum):

$$f(x)=\sum i=1n[x_{i2}-10cos(2\pi xi)+10] \quad (9)$$

Rosenbrock function (ill-conditioned function, used to test the robustness of algorithms in nonlinear and ill-conditioned terrains):

$$f(x)=\sum i=1n-1[100(x_i+1-x_{i2})2+(x_i-1)2] \quad (10)$$

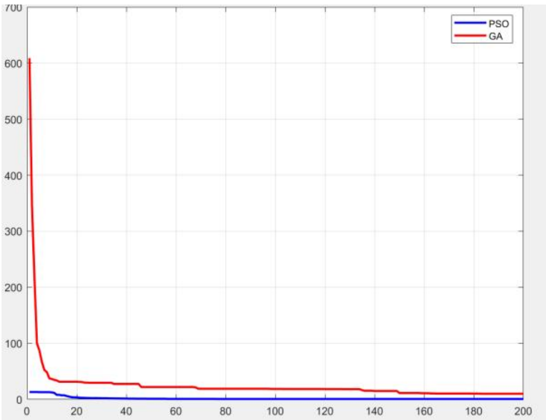
Schwefel function (used to test the performance when the global extremum is located at the edge of the search space):

$$f(x)=418.9829n-\sum i=1nxisin(\mid x_i \mid) \quad (11)$$

As shown in the convergence speed comparison data in **Table 1**, PSO converges significantly faster than GA in both single-peak problems and low-dimensional multi-peak problems. This is because PSO has a direct position update mechanism, while GA needs to gradually optimize through complex crossover and mutation operations.

Figure 5

Convergence speed comparison curve



As shown in the convergence speed comparison data in **Table 2**, PSO converges significantly faster than GA in both single-peak problems and low-dimensional multi-peak problems. This is because PSO has a direct position update mechanism, while GA needs to gradually optimize through complex crossover and mutation operations.

Table 2

Comparison Data of Convergence Speed

Function type	PSO (number of iterations)	GA (Number of Iterations)
Sphere (10D)	35 ± 2.1	62 ± 3.8
Rastrigin (10D)	120 ± 15	180 ± 20

3.2. Fitness comparison

The comparison data in **Table 3** regarding the quality of the solution represents the average value and standard deviation of the final fitness after 30 runs. In multi-modal and high-dimensional problems, the solutions obtained by GA are of higher quality (closer to the global optimum) because the mutation operation of GA can escape from local optima; while PSO is easily influenced by the initial particle distribution and may get trapped in local optima (Gao, 2025; Kheirdast et al., 2024).

Table 3

Comparative Data on Quality of the Table

Algorithm	Optimal fitness mean	Standard deviation
PSO	48.3	± 5.2
GA	15.7	± 2.1

4. Application scenario

As shown in **Table 4** for scenario selection, PSO and GA have different application scenarios and possess their own advantages and disadvantages. The choice should be made based on the problem type, computing resources, and quality requirements of the solution. New technologies such as hybrid algorithm development, multimodal optimization, and quantum computing can be integrated (Anwar and Vilas).

Table 4

Scene Selection

Algorithm selection	Priority scenario	Application scenario
PSO	Continuous space, real-time dynamics, high-dimensional parameter optimization	Discrete combinatorial problem (requires encoding conversion, efficiency decreases)
GA	Discrete combinatorics, multi-objective, structural topology optimization	Large-scale continuous problems (with high computational cost)

PSO and GA have extensive application fields. PSO can quickly and accurately determine the parameters of circuit components in circuit design, optimize the design of components in mechanical manufacturing, and improve the convergence speed of Convolutional Neural Network. It is also commercially applied in the autonomous navigation system of agricultural drones. GA is crucial for comparing and optimizing gene sequences in bioinformatics. In the logistics field, it is responsible for planning delivery routes and solving vehicle routing problems. In industrial production scheduling, it optimizes processes and task allocation based on various factors to improve efficiency (Gyan and Amit, 2023).

5. Summary and Outlook

This article analyzes and explains two classic swarm intelligence optimization algorithms, namely PSO (PSO) and GA (GA). It also introduces their principles. Through data analysis, it is recognized that PSO converges faster in single-peak and low-dimensional multi-peak problems, and is more suitable for continuous-time scenarios. While GA performs better in multi-peak and high-dimensional solution quality, and is more suitable for discrete combination and other scenarios.

5.1. The Implications of Learning and Practice

Through the analysis of various experiments, we have mastered the algorithm learning framework of “principal analysis - performance verification - scenario adaptation”, which can solve problems from both the mathematical model and data analysis perspectives. By comparing the convergence speed and solution quality of the PSO algorithm and the GA, we deeply understood that “there is no universal algorithm”, and more often, a combination of the two is needed to achieve the desired effect.

5.2. Future Outlook

In the future, the PSO algorithm and the GA are likely to be deeply integrated. The two complement each other and make up for each other’s weaknesses. PSO converges quickly, while GA has a strong ability for global search. When combined, they can enable many fields of society to solve problems that were previously unsolvable and move towards the path of progress.

Conflict of interest: The authors declare no conflict of interest.

Funding: This research received no external funding.

References

[1] Abibou, S., Bourakadi, E. D., & Yahyaouy, A. (2024). Optimizing hydrogen refueling station recommendations: A comparative analysis between GA and PSO. *SN Computer Science*, 5(8), 991. https://doi.org/10.1007/s42979-024-02946-7

[2] Anwar, T. Q., & Vilas, W. (2023). Wind farm layout optimization through optimal wind turbine placement using a hybrid PSO and GA. *Environmental Science and Pollution Research*, 30(31), 77436–77452. https://doi.org/10.1007/s11356-023-27901-6

[3] Deng, L., & Liu, S. (2025). Collective dynamics of particle swarm optimization: A network science perspective. *Physica A: Statistical Mechanics and its Applications*, 675, 130778.

[4] Gao, W. (2025). Application of improved particle swarm optimization algorithm combined with genetic algorithm in shear wall design. *Systems and Soft Computing*, 7, 200198.

[5] Gupta, N. S., & Chak, K. S. (2024). Experimental investigations and optimization of surface roughness using response surface methodology coupled with GA and PSO techniques in grinding of Inconel 718. *International Journal of Precision Engineering and Manufacturing*, 25(12), 2437–2453. https://doi.org/10.1007/s12541-024-00998-6

[6] Gyan, S., & K., A. C. (2023). Hybrid modified PSO with GA based workflow scheduling in cloud-fog environment for multi-objective optimization. *Cluster Computing*, 27(2), 1947–1964. https://doi.org/10.1007/s10586-023-04033-7

[7] Hu, W., Zhang, Y., Liu, L., et al. (2024). Study on multi-objective optimization of construction project based on improved GA and PSO. *Processes*, 12(8), 1737. https://doi.org/10.3390/pr12081737

[8] Huber, K., Wirtz, T., & Hoang, Q. H. (2024). CPOpt: A modular framework for GA optimization and post-optimization analysis in complex charged particle optical design. *Nuclear Instruments and Methods in Physics Research, Section A*, 1067, 169702.

[9] Jin, Y., Xuerong, C., & Juan, L. (2021). Particle filter algorithm optimized by genetic algorithm combined with particle swarm optimization. *Procedia Computer Science*, 187, 206–211. https://doi.org/10.1016/j.procs.2021.04.044

[10] Kheirdast, A., Jozi, A. S., & Rezaian, S. (2024). Comparing the performance of

GA and PSO algorithm in allocating and scheduling fire stations. *International Journal of Environmental Science and Technology*, 22(1), 445–458. https://doi.org/10.1007/s13762-023-04957-y

[11] Laishram, A., Padmanabhan, V., & Lal, P. R. (2018). Analysis of similarity measures in user-item subgroup based collaborative filtering via genetic algorithm. *International Journal of Information Technology*, 10(4), 523–527. https://doi.org/10.1007/s41870-018-0157-5

[12] P, G. M. S., K, N. H. N., Jain, K. A., et al. (2025). Convergence of improved PSO based ensemble model and explainable AI for accurate detection of food adulteration in red chilli powder. *Journal of Food Composition and Analysis*, 143, 107577.

[13] Said, G. A., & Jawale, D. (2025). Design of an iterative method for enhanced disease detection and feature selection using particle swarm optimization, genetic algorithms, and convolutional neural networks. *Evolving Systems*, 16(3), 95.

[14] Saini, M., Sinwar, D., Swarith, M. A., et al. (2023). Reliability and maintainability optimization of load haul dump machines using GA and PSO. *Journal of Quality in Maintenance Engineering*, 29(2), 356–376. https://doi.org/10.1108/JQME-06-2021-0045

[15] Seghier, B. A. E. M., Carvalho, H., & Keshtegar, B. (2020). Novel hybridized adaptive neuro-fuzzy inference system models based particle swarm optimization and genetic algorithms for accurate prediction of stress intensity factor. *Fatigue & Fracture of Engineering Materials & Structures*, 43(11), 2653–2667. https://doi.org/10.1111/ffe.13325

[16] Shui, P., Zhang, Z., Xie, C., et al. (2025). Optimization of environmental conditioning equipment based on PSO algorithm and GA. *Frontiers in Computing and Intelligent Systems*, 11(1), 85–88.

[17] Gyan, S., & K., A. C. (2023). Hybrid modified PSO with GA (GA) based workflow scheduling in cloud-fog environment for multi-objective optimization. *Cluster Computing*, 27(2), 1947–1964. https://doi.org/10.1007/s10586-023-04033-7