*Journal of Network Intelligence*
**Taiwan Ubiquitous Information**
 ISSN 2414-8105
Volume **10**, Number **1**, February **2025**

# The Application of Bio-Inspired Optimization Algorithms in Structural Optimization Mathematical Models

Ji Liu

Department of Basic Courses
Hebei Vocational University of Industry and Technology, Shijiazhuang 050091, P. R. China
15933117159@163.com

Wen-Tao Wang*

Academic Affairs Office
Hebei Vocational University of Industry and Technology, Shijiazhuang 050091, P. R. China
wangwentao401@163.com

Yang Liu

College of Management Science
University of Cyberjaya, Cyberjaya 63000, Malaysia
et4601@163.com

*Corresponding author: Wen-Tao Wang
Received January 27, 2024, revised May 3, 2024, accepted July 19, 2024.

---

Abstract. *With the increasing prevalence of bio-inspired optimization algorithms and advancements in computing power, researchers are increasingly adopting these algorithms to tackle structural optimization problems. The core concept involves transforming structural optimization problems into mathematical expressions and utilizing bio-inspired optimization algorithms to efficiently search the solution space, thereby enhancing the overall effectiveness and quality of optimization results. Structural optimization is a pivotal technology aimed at adjusting the design parameters of a structure to maximize its performance within specific constraints. Nevertheless, traditional optimization algorithms encounter limitations when dealing with structural optimization, such as susceptibility to local optima and high computational complexity. Consequently, this paper explores a novel approach by leveraging bio-inspired optimization algorithms to overcome these limitations and improve the effectiveness of solving structural optimization problems. The research presented herein primarily focuses on establishing a mathematical model that captures the essence of the structural optimization problem and employs bio-inspired optimization algorithms to search for optimal solutions within the solution space. These algorithms simulate biological behaviors and evolutionary processes, endowing them with global search capabilities and robustness. To validate the applicability of bio-inspired optimization algorithms in structural optimization, a series of numerical experiments and comparative analyses are performed using real-world structural problems as benchmarks. By comparing bio-inspired optimization algorithms with traditional approaches like gradient-based methods and constraint optimization methods, this paper demonstrates the advantages and effectiveness of bio-inspired optimization algorithms in solving structural optimization problems. The experimental results underscore the ability of bio-inspired optimization algorithms to obtain superior structural design solutions under comparable computational complexities.*
**Keywords:** Bio-inspired optimization algorithm; structural optimization problem; mathematical model; search space; optimal solution

1. **Introduction.** Optimization problems are fundamental challenges in scientific research and engineering computation. They involve finding parameter values that optimize performance metrics while satisfying constraints. These problems have broad applications in management, economics, society, and engineering, significantly impacting our daily lives [1]. However, traditional mathematical optimization methods face limitations in tackling complex problems, including dimensionality, local optima, and slow convergence rates.

Luckily, bio-inspired optimization algorithms have emerged as promising solutions. These algorithms simulate behaviors and thinking structures observed in biological evolution, offering new approaches to solving complex optimization problems [2]. They draw inspiration from the evolutionary processes of biological populations, aiding the exploration of solution spaces. This research aims to investigate the application of bio-inspired optimization algorithms in mathematical models for structural optimization. Structural optimization problems are crucial across engineering fields such as architecture, aerospace, automotive, and materials. By optimizing design parameters, structures' performance can be enhanced while meeting design constraints. However, traditional optimization methods often struggle to effectively address structural optimization problems. This paper investigates the use of bio-inspired optimization algorithms, specifically emphasizing the PSO algorithm. PSO is an optimization technique inspired by the collective behavior of particles, initially observed in the foraging behavior of birds [3]. It mimics information exchange and collaboration among individuals to iteratively adapt parameter values in pursuit of the best possible solution. PSO offers advantages like global search capability, ease of implementation, and fast convergence, making it widely utilized in structural optimization [4].

This paper delves into the principles and key parameters of the PSO algorithm, validating its effectiveness in structural optimization mathematical models through illustrative examples. By studying PSO, we aim to enhance the efficiency and quality of solving structural optimization problems, providing valuable guidance for decision-makers and researchers in the engineering field.

1.1. **Related Work.** Bio-inspired algorithms are computational techniques and methods that mimic the structural characteristics, evolutionary patterns, behavior, and thinking processes observed in humans, nature, and other biological populations. A variety of optimization problems can be addressed through the utilization of diverse algorithms. Genetic Algorithms, Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Artificial Fish Swarm Algorithm (AFSA) are renowned bio-inspired optimization techniques, exhibiting distinct advantages and finding applications in numerous domains [5]. This study specifically focuses on harnessing the power of the PSO algorithm to optimize mathematical models that involve structural components.

Particle Swarm Optimization, is a swarm intelligence algorithm devised by Kennedy and Eberhart in the early 1990s. Inspired by the foraging behavior of bird flocks and building upon Heppner's bird model research, the PSO algorithm was developed. Since its initial introduction, researchers have made significant efforts to enhance the algorithm's structure and performance [6]. Mirjalili, and Lewis [7] introduced a discrete binary PSO algorithm specifically designed for effectively solving discrete problems. Yang et al. [8] introduced the concept of inertia weight as a parameter to regulate the momentum of particles from previous iterations. The introduction of this parameter added a new level of flexibility and adaptability to the PSO algorithm. Jin and Rahmat-Samii [9] expanded the application of PSO to address Multi-objective Optimization Problems (MOPs). This extension showcased how the algorithm effectively tackles intricate optimization problems with multiple conflicting objectives. Overall, the continuous efforts and advancements in PSO algorithm research have contributed to its versatility and applicability in various optimization scenarios. Liu et al. [10] further extended PSO by introducing the Pareto dominance concept to guide particles' search direction and preserving non-dominated particles in a global external archive to guide other particles in subsequent iterations. Kiranyaz et al. [11] proposed an adaptive dynamic environment PSO algorithm, which allows particles to reset their best positions during the iteration process to prevent stagnation caused by historical experience. Moradi, and Abedinie [12] conducted additional research on PSO, resulting in an algorithm that guarantees global convergence. Cesare et al. [13] conducted in-depth research on the topology and information transmission mechanisms of PSO and proposed a holographic particle swarm optimization algorithm.

Intelligent optimization algorithms have demonstrated remarkable performance in solving complex mathematical models for structural optimization, leading to their increasing application across various industries. To illustrate their efficacy, let's consider the optimization of truss structure models. Cao et al. [14] proposed an enhanced Teaching-Learning-Based Optimization (TLBO) algorithm for dimension optimization of truss structures. Their approach included effectively handling constraints within the algorithm and mapping all feasible solutions on the boundary of the feasible domain. This improvement resulted in reduced structural analysis complexity and improved convergence speed. Kaveh and Talatahari [15] employed the Ant Colony Optimization (ACO) algorithm to optimize spatial truss structures. They transformed the discrete variable truss design problem into a variant of the Traveling Salesman Problem (TSP). By utilizing the path length obtained from the TSP as a metric for truss optimization, they achieved favorable optimization outcomes. Degertekinet al. [16] utilized the Subset Simulation algorithm to optimize discrete variable truss structures. They compared its performance with other classical optimization algorithms, effectively demonstrating its efficacy. Kaveh and Ghazaan [17] proposed an enhanced Collision-Based Optimization (CBO) algorithm tailored for weight optimization of truss structures under frequency constraints. The algorithm's effectiveness was validated through a series of simulation experiments. Another study, conducted by Dimou and Koumousis [18], utilized the Particle Swarm Optimization algorithm to optimize reliability-based truss structures. In their approach, random particles within the PSO algorithm were considered as external forces leading to structural yielding, highlighting the robustness of the algorithm. The simulation results provided convincing evidence of the Particle Swarm Optimization algorithm's effectiveness in reliability-based optimization of truss structures.

Overall, these pioneering studies solidify the role of intelligent optimization algorithms in enhancing the performance of truss structures across a range of optimization scenarios.

1.2. **Motivation and contribution.** Structural optimization poses a significant challenge in achieving optimal designs for structures. Traditional optimization methods have limitations when confronted with complex structural optimization problems. Consequently, researchers are actively seeking more effective optimization algorithms to enhance structural design performance. This study aims to investigate and utilize PSO, an optimization algorithm inspired by nature, to tackle the optimization problem linked to truss structure models.

The paper's contributions are as follows:

(1) To address the problem of traditional multi-objective optimization algorithms struggling to preserve multiple Pareto optimal solution sets, this study proposes an innovative multi-modal multi-objective particle swarm optimization (MPSO) algorithm using

a circular topology structure and neighborhood perturbation strategy. By employing an index-based non-overlapping circular topology structure, without specifying any niche parameter, the algorithm encourages the formation of multiple independent search niches within the population, enabling it to discover a greater number of optimal solutions.

(2) This study introduces a novel application of bio-inspired optimization algorithms to tackle multi-objective optimization problems and presents an automatic switching mechanism between global search and local search modes to strike a balance. The mechanism dynamically adjusts the algorithm's search strategy during the optimization process, ensuring convergence while maintaining diversity.

(3) In order to enhance the algorithm's capability to search for more optimal solutions, a stagnation detection strategy is introduced in this paper to perturb the neighborhood best particles, increasing the diversity of the particle swarm and preventing premature convergence to a particular Pareto optimal solution set. This strategy helps the algorithm escape local optima and further improves its global search ability.

## 2. Relevant theoretical analysis.

### 2.1. Particle Swarm Optimization algorithm.

The PSO algorithm draws inspiration from birds' collective behaviors, such as foraging and migration, and demonstrates remarkable effectiveness in tackling intricate optimization problems. In the PSO algorithm, a predetermined number of particles are initially positioned randomly within the feasible region of the problem space. Each particle adjusts its state by continuously updating its velocity, guided by the best position it has encountered personally and the globally best position identified across the entire population. This adaptive process allows the particles to explore and converge towards better regions in search of optimal solutions. In the PSO algorithm, the search space is represented as $D$-dimensional, with a total of $N_s$ particles. For a given particle at the $t$-th iteration, its velocity and position are denoted as $v_i^t = (v_{i,1}^t, v_{i,2}^t, \ldots, v_{i,D}^t)$ and $\mathrm{x}_i^t = \left(x_{i,1}^t, x_{i,2}^t, \ldots, x_{i,D}^t\right)$ respectively. The best position found by that particle up to the current iteration is represented as $p_i^t = (p_{i,1}^t, p_{i,2}^t, \ldots, p_{i,D}^t)$, which is the individual best. The optimal position discovered thus far by the entire population during the $t$-th iteration is represented as $g^t = (g_1^t, g_2^t, \ldots, g_D^t)$, representing the global best solution encountered. In each $(t+1)$-th iteration, each particle in the swarm recalibrates its velocity and adjusts its position within the search space. These updates are influenced by the particle's individual experiences as well as the collective behavior of the entire swarm. These updates, including the inertia weight scheme, are defined by Equations (1) and (2) as follows:

$$v_{i,j}^{t+1} = \omega v_{i,j}^t + c_1 r_1 (p_{i,j}^t - x_{i,j}^t) + c_2 r_2 (g_j^t - x_{i,j}^t) \tag{1}$$

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1} \tag{2}$$

Here, $\omega$ represents the inertia weight that controls the particle's current momentum, $c_1$ and $c_2$ are acceleration coefficients determining the influences of the individual and global best positions on the particles' adjustments, $r_1$ and $r_2$ are randomly generated values between 0 and 1 used to introduce stochasticity in the algorithm. Its position update equation remains unchanged, while the velocity update equation is as follows:

$$v_i^{t+1} = \omega v_i^t + c r_1 (pbest_i^t - X_i^t) + c_2 r_2 (nbest_i^t - X_i^t) \tag{3}$$

Equation (1) defines the velocity update mechanism in the PSO algorithm, comprising of three key components. The initial step entails the product of the inertia weight and the particle's existing velocity, governing the impact of the particle's current velocity on its movement direction. This component helps balance local exploitation and global exploration. The second component is the particle's self-awareness, representing the particle's memory of its own experience. Its purpose is to preserve diversity among the swarm members and avoid the algorithm becoming trapped in local optima. The third component is the social awareness, representing the sharing of information and cooperative behavior among particles. It allows particles to benefit from the collective experience of the swarm, leading to improved convergence speed of the algorithm.

The PSO algorithm finds extensive use in fields like engineering due to its inherent advantage of having a concise parameter configuration. However, determining the optimal parameters becomes challenging as they are random variables and lack clear theoretical guidance. Researchers often resort to trial and error to adjust parameter values when solving different optimization problems, leading to increased experimental workload. Therefore, parameter research in PSO holds significant theoretical importance. As an example, let's consider the introduction of a linearly decreasing inertia weight strategy. In the early stages of evolution, a larger inertia weight enables particles to possess strong global exploration capabilities, extensively searching the solution space to discover new regions. During the iteration process, the inertia weight gradually decreases, enabling particles in the later stages to perform intricate exploration around the optimal solution and attain enhanced precision. This decreasing inertia weight strategy can be mathematically represented by Equation (4).

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{T_{\max}} \times t \tag{4}$$

Here, $\omega_{\max}$ and $\omega_{\min}$ correspond to the upper and lower bounds for the inertia weight. On the other hand, $T_{\max}$ denotes the maximum number of iterations, and $t$ denotes the current iteration count.

Here the flowchart of the PSO algorithm is shown as Figure 1.

2.2. **Mathematical model.** In practical structural optimization design, applying mathematical optimization methods involves the following steps: First, a mathematical model is constructed, followed by structural analysis and model validation to address real-world engineering optimization problems. Expressing the structural optimization design problem in mathematical language is a crucial step, as it forms the basis for establishing the mathematical model of structural optimization. The constructed mathematical model must accurately reflect the actual loading conditions experienced by the engineering structure under optimization. Thus, constructing an appropriate mathematical model stands as one of the most pivotal steps in solving optimization problems. When dealing with practical engineering structural optimization problems, the number of design variables is constrained by the dimensionality of the optimization model. Assuming there are $n$ design variables, these variables can be effectively represented as a vector, with each coordinate in the vector serving as an individual element and a distinguishing factor for different optimization strategies. Consequently, the optimization problem for the $n$ design variables can be expressed using Equation (5):

$$x = [x_1, x_2, \ldots, x_n]^T \tag{5}$$

In practical engineering optimization, design variables must conform to specific rules called constraints. These constraints cover a range of requirements including local stability, frequency, stiffness, strength, and other factors that ensure the optimized structure functions effectively. Additionally, constraints include design specifications and operational requirements for the structure's usage. In the mathematical model of structural

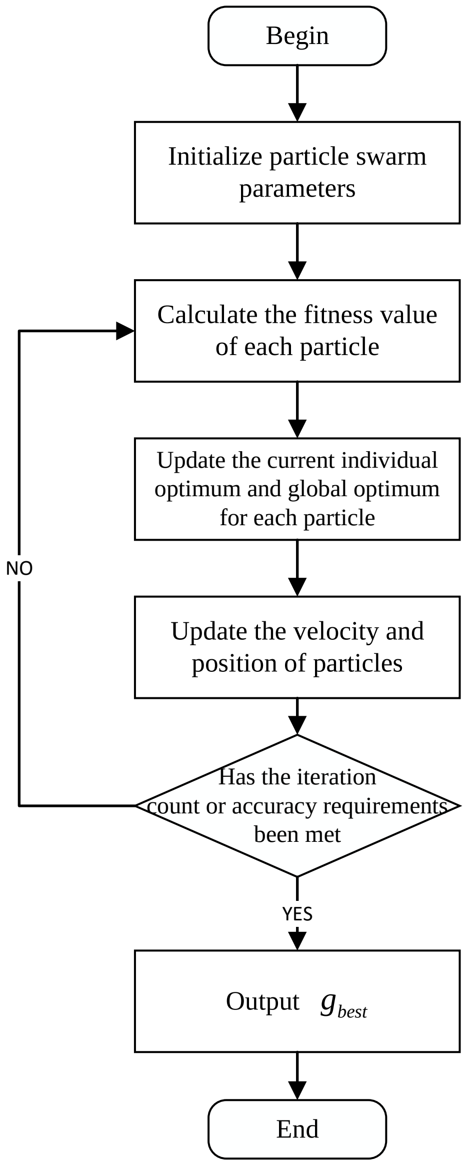


Figure 1. Flowchart of Particle Swarm Optimization Algorithm

optimization design, constraints are divided into two forms: equality constraints and inequality constraints. The mathematical expressions are as follows:

$$g_i(x) \leq 0, \quad i = 1, 2, \ldots, p \tag{6}$$

$$h_j(x) = 0, \quad j = 1, 2, \ldots, q \tag{7}$$

Moreover, we can employ an objective function that integrates the design variables to articulate the desired goals. The numerical output of the objective function serves as a direct measure for evaluating the effectiveness of design solutions. Thus, the careful selection of a suitable objective function holds paramount importance in achieving successful structural optimization design. In the case of optimizing a practical structure with $n$ design variables, the objective function can be mathematically represented as follows:

$$f(x) = f(x_1, x_2, \ldots, x_n) \tag{8}$$

In summary, taking the objective function minimization as an illustration, the optimization problem for an engineering structure can be formulated using the following mathematical expression:

$$\begin{cases} \min f(x) \\ \text{s.t. } g_i(x) \leq 0, \quad i = 1, 2, \ldots, p \\ h_j(x) = 0, \quad j = 1, 2, \ldots, q \end{cases} \tag{9}$$

For a practical engineering structural optimization problem, the optimal solution is obtained when the objective function $f(x)$ is minimized while satisfying the structural constraints. In this case, the design variables $x = [x_1, x_2, \ldots, x_n]^T$ obtained through the search represent the optimal solution to the problem.

## 3. Multimodal multi-objective optimization algorithm based on ring topology and neighborhood perturbation.

3.1. **Ring Topology.** The ring topology in PSO has proven to be effective in controlling information propagation speed and algorithm convergence rate, as shown in Figure 2. In this topology, each particle communicates with its two neighboring particles and updates its position based on the best position within its local neighborhood. The local PSO within the ring topology encourages stable niching behaviors, leading to the discovery of more optimal solutions. In order to promote population diversity, this study employs a non-overlapping ring topology based on indices to enhance the algorithm's exploration ability in seeking optimal solutions. For instance, in Figure 2, let's consider a particle swarm with 10 particles. These particles form a ring structure based on their indices, with every three adjacent particles forming a neighborhood. For example, particles with indices 1, 2, and 3 are neighbors, particles with indices 4, 5, and 6 are neighbors, and so on. In this way, multiple parallel niches are formed. Compared to PSO with overlapping neighborhood structures, the niches in the non-overlapping ring topology are independent of each other. This allows them to conduct detailed local searches within their respective regions, avoiding interference between different optimal solutions in different neighborhoods. Consequently, the non-overlapping ring topology further enhances population diversity and search accuracy.

3.2. **Introduction of Conversion Probability.** Balancing global exploration and local exploitation becomes more intricate due to the mapping between the decision space and the objective space. To address this, this paper draws inspiration from the concept of global and local pollination transformation probabilities in the Flower Pollination Algorithm. A higher transformation probability improves the algorithm's global search capability and population diversity but may reduce search accuracy. In contrast, a reduced transformation probability allows for finer adjustments at the local level but runs the risk of getting stuck in local optima.

To address the trade-off between global and local search, this research introduces a linearly decreasing transformation probability approach. During the algorithm iteration, the particle's position update strategy adjusts the search mechanism between global and local search. In the initial search stages, a larger transformation probability, denoted as $\rho$, is used to explore the search space extensively. In later iterations, a smaller transformation probability, $\rho$, is employed to thoroughly search a specific region and achieve higher precision solutions. This approach effectively addresses the challenge of balancing the algorithm's exploration and exploitation capabilities. The adjustment formula for the transformation probability, $\rho$, can be expressed as:

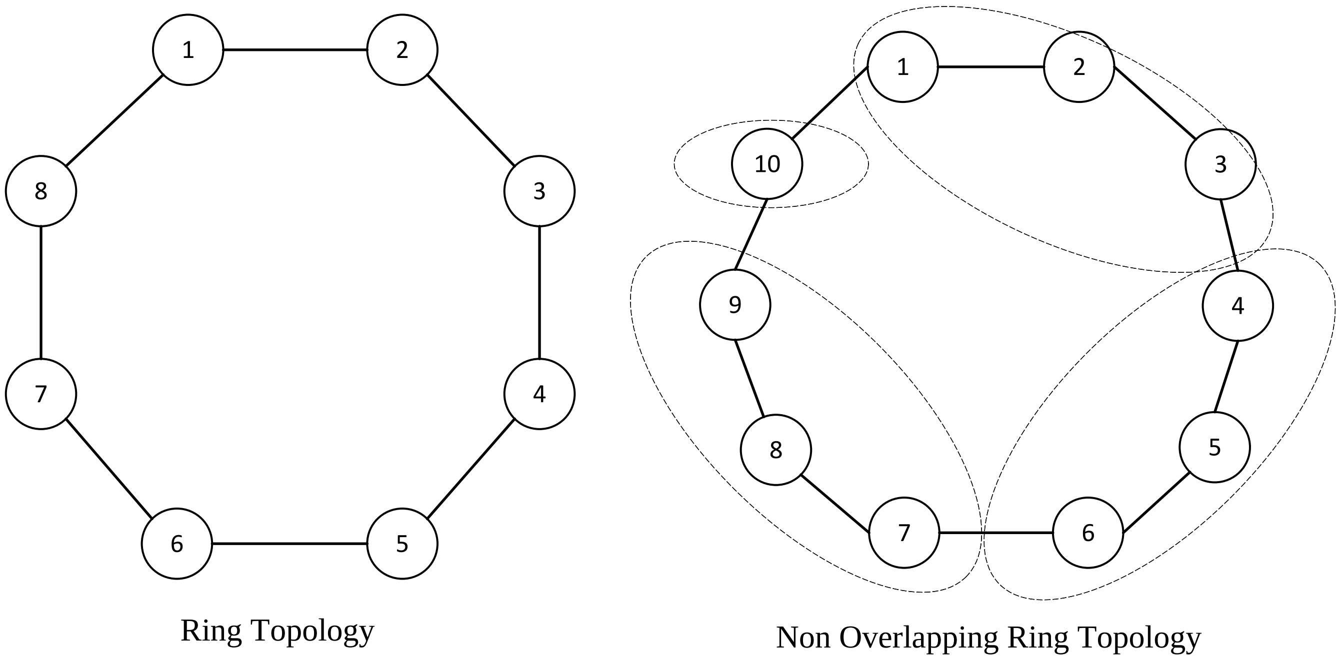


Figure 2. Topology Structure

$$\rho = \rho_{\max} - \frac{\rho_{\max} - \rho_{\min}}{t_{\max}} t \tag{10}$$

Here, $\rho_{\max}$ and $\rho_{\min}$ represent the maximum and minimum transformation probabilities, typically set to 0.95 and 0.4, respectively. $t_{\max}$ and $t$ denote the maximum number of iterations and the current iteration count, respectively.

3.3. **Stagnation detection strategy.** The local pattern PSO weakens the influence of the global best particle during the evolution process, where a particle's state is led by its best neighbor particle. Once a particle gets trapped in a local optimum, it may cause other individuals in the same neighborhood to stop moving and remain stagnant in the search space. To guide particles to escape local optima and prevent a decline in optimization performance, a stagnation detection strategy is introduced. The strategy utilizes local search methods to improve the quality of obtained solutions. Before each particle update, the current best neighbor $nbest_i^t$ is compared with the previous generation's best neighbor $nbest_i^{t-1}$. If the current best neighbor $nbest_i^t$ is better than the previous best neighbor $nbest_i^{t-1}$, the stagnation factor $\zeta_i$ of the $i$-th particle is set to zero. If it does not show improvement, $\zeta_i$ is increased by 1. The formula is shown as follows:

$$\zeta_i = \begin{cases} \zeta_i + 1, & nbest_i^t = nbest_i^{t-1} \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

When the local optimum does not improve for $N$ consecutive iterations ($\zeta_i \geq N$), it indicates that the algorithm may be in a stagnant state. To avoid this phenomenon, a Gaussian disturbance strategy is applied to the best neighbor particle $nbest_i^t$, resulting in a new local optimum position $nbest_i^t$. The update equation is given as follows:

$$nbest_i^t = nbest_i^t (1 + G(\sigma)) \tag{12}$$

where $G(\sigma)$ represents a random number that follows a Gaussian distribution.

4. **Experiments and analysis of results.** To validate the efficacy of the proposed modified particle swarm optimization (MPSO) algorithm, two sets of experiments were performed. In the first set, we compared PSO algorithms using overlapping and non-overlapping ring topologies to assess the performance improvement of the non-overlapping ring topology. In the second set, MPSO was benchmarked against four established multi-modal multi-objective optimization algorithms to evaluate its performance. All simulations in this section were implemented on a computer using MATLAB 2017b software.

4.1. **Performance indicators.** This paper uses two metrics, PSP (Pareto Set Proximity) and HV (Hypervolume), to evaluate the algorithm performance. PSP is employed to measure the resemblance between the obtained Pareto set and the true Pareto Set (PS). A higher $PSP$ value signifies a closer proximity of the algorithm's PS to the true PS and signifies a superior distribution. The calculation formula for $PSP$ is shown below:

$$PSP = \frac{CR}{IGDX} \tag{13}$$

To measure the overall performance of each $PSP$ algorithm, a performance score is introduced for ranking. Let there be $l$ algorithms: $Alg_1, Alg_2, \ldots, Alg_l$. If $Alg_j$ exhibits significantly better performance in terms of the $PSP$ measure compared to $Alg_i$, the score $\delta_{i,j}$ is assigned as 1; otherwise, it is assigned as 0. The calculation formula for the performance score of $Alg_i$ is depicted in Equation (14).

$$P(Alg_i) = \sum_{j=1, j\neq i}^{I} \delta_i \tag{14}$$

Among them, $CR$ represents the coverage rate of the obtained PS compared to the true PS. The indicator $IGDX$ (Inverted Generational Distance) assesses the diversity and convergence of solutions in the decision space. The calculation formula for $IGDX$ is shown below:

$$IGDX(O, P^*) = \frac{\sum_{v\in P^*} d(v, O)}{|P^*|} \tag{15}$$

Here, $O$ denotes the Pareto set acquired through the multi-modal algorithm, $P^*$ signifies a set of reference points that are uniformly distributed across the true Pareto set. The term $d(v, O)$ denotes the minimum Euclidean distance between a point $v$ in the reference set and a point in the solution set $O$.

The $HV$ indicator represents the size of the hypervolume enclosed by the Pareto Front (PF) obtained by the algorithm and a set of reference points. $HV$ can effectively assess both convergence and diversity of the algorithm, with a larger $HV$ value indicating superior overall performance.

4.2. **Comparison between overlapping and non-overlapping ring topologies.** To assess the algorithm's efficacy with a non-overlapping ring topology, a comparative analysis was conducted using five multi-modal multi-objective test functions (MMF1, MMF3, SYM-PART simple, SYM-PART rotated, Omni-test). The comparison was made between multi-objective PSO algorithms employing overlapping and non-overlapping ring topologies. Except for the difference in particle swarm topology, the rest of the algorithm remained the same.

Figure 3 shows the average best individual fitness values ($PSP$) of the two algorithms on each test function and displays the Pareto optimal solutions they obtained on MMF3.

To facilitate the description, we will refer to the algorithm with overlapping ring topology and the one with non-overlapping ring topology as MPSO-o and MPSO-n, respectively. The results indicate that MPSO-n achieved better $PSP$ values than MPSO-o on all test functions except MMF1. Additionally, on the Omni-test, MPSO-n obtained a more uniform and widely distributed set of optimal solutions, indicating that the solution set of MPSO-n is closer to the true Pareto optimal set and exhibits better diversity. Therefore, it can be inferred that the non-overlapping ring topology is effective in improving the algorithm's performance. This improvement can be attributed to the independent and parallel local search in each small habitat of the non-overlapping ring topology. This structure prevents certain non-dominated solutions from quickly dominating the entire population and enables a finer search in the region where the optimal solutions are located. Therefore, employing a non-overlapping ring topology demonstrates a beneficial effect on enhancing the performance of the algorithm in multi-objective optimization scenarios.

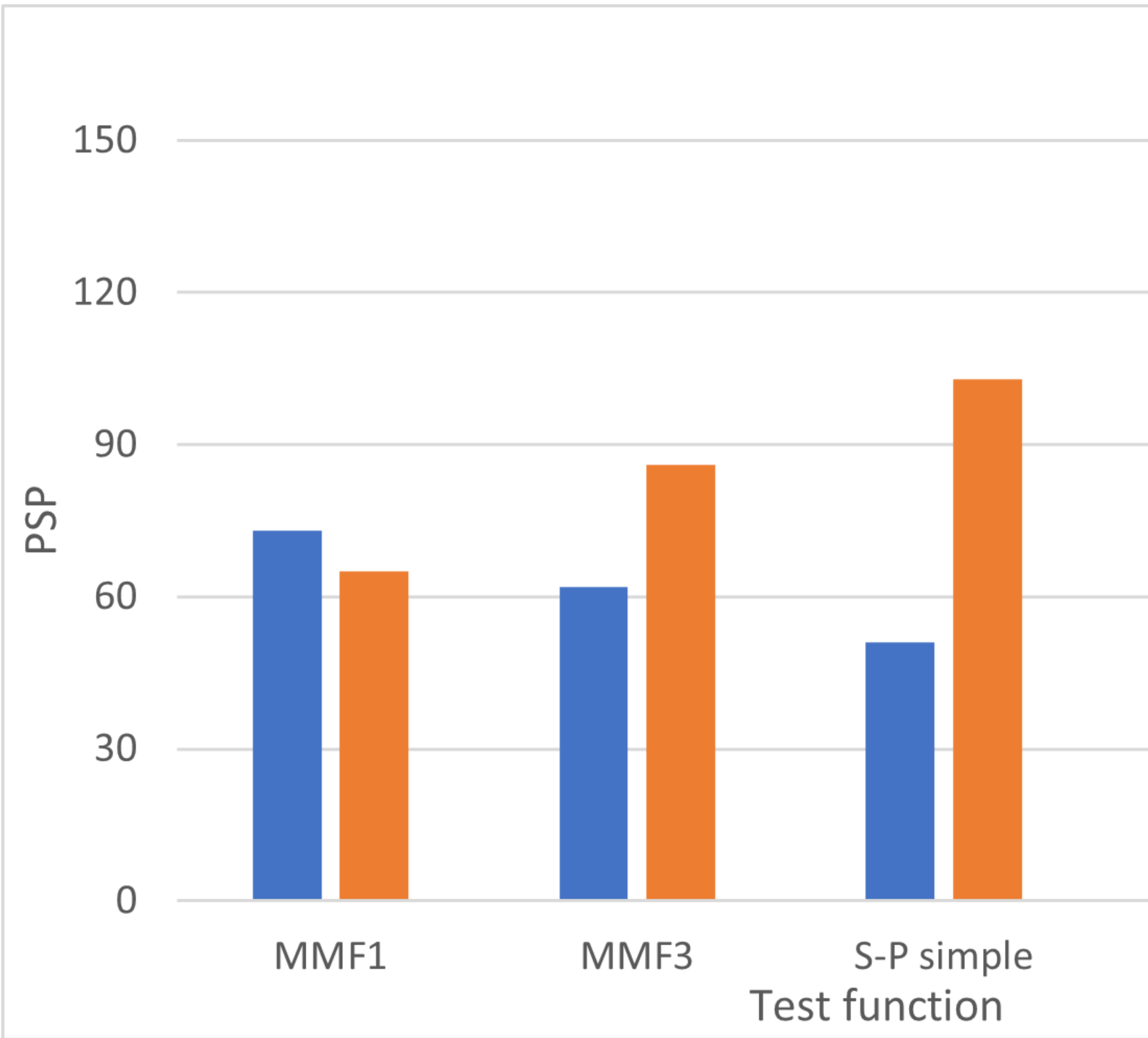


Figure 3. $PSP$ mean obtained by two algorithms

4.3. **Baseline comparison.** The performance of the proposed MPSO algorithm was evaluated by comparing it to other algorithms, namely Omni-optimizer, MO-Ring-PSO-SCD, DN-NSGA-II, and TriMOEA-TA&R. These four algorithms are existing algorithms known for effectively handling multi-modal optimization problems. All algorithms were set with a maximum evaluation count of 80000 and a population size of 800. Each algorithm was independently run 30 times.

Table 1. The average and standard deviation of *PSP* measures for 5 algorithms

| **Test function** | **Algorithm** | | | | |
|---|---|---|---|---|---|
| | Omni-optimizer | MO-Ring-PSO-SCD | DN-NSGA-II | TriMOEA-TA&R | MPSO |
| MMF1 | 46.18±6.05 | 59.37±2.80 | 38.41±5.05 | 8.74±2.63 | 68.61±2.03 |
| MMF3 | 68.95±40.25 | 124.16±10.75 | 63.23±21.55 | 57.57±16.18 | 129.00±13.75 |
| SYM-PART simple | 0.85±4.03 | 9.39±1.56 | 0.69±0.93 | 18.36±2.78 | 9.46±1.04 |
| SYM-PART rotated | 2.88±3.41 | 10.06±1.28 | 5.53±6.88 | 8.33±5.65 | 14.15±0.86 |
| Omni-test | 0.72±0.37 | 8.95±0.56 | 1.76±0.30 | 13.72±0.43 | 8.54±0.13 |

Tables 1 and 2 provide the average and standard deviation values of the PSP measure and HV measure for the five algorithms on each test function. It is observed from Table 1 that MPSO consistently achieved the highest PSP values across all five functions. This indicates that the proposed algorithm generated solutions that were closer to the true Pareto set and had better distribution in most test functions. Notably, TriMOEA-TA&R exhibited the highest average PSP value for SYM-PART simple and Omni-test, followed by MPSO and MO-Ring-PSO-SCD. This could be attributed to TriMOEA-TA&R incorporating distance-related and position-related variables while maintaining diversity in both the objective and decision spaces. Examining the data in Table 2, it can be observed that the five algorithms obtained comparable HV values for each test problem. MPSO achieved the highest HV value for the MMF1 test function, closely followed by Omni-optimizer. Omni-optimizer exhibited the highest average HV across multiple test functions. DN-NSGA-II performed well for SYM-PART simple, SYM-PART rotated, and Omni-test. Although MPSO had slightly lower HV values compared to Omni-optimizer and DN-NSGA-II, the difference was minimal. Given that emphasizing a satisfactory spread of solutions in the decision space can impact their distribution in the objective space, a marginal decrease in HV values is considered acceptable.

In conclusion, the proposed MPSO algorithm in this paper achieves high PSP values and acceptable HV values when dealing with mathematical model structural optimization. MPSO achieves a desirable equilibrium between the dispersion of solutions in the decision space and the objective space. When comparing its performance to other algorithms, it becomes clear that MPSO can simultaneously maintain diversity and convergence in the objective space, ensuring the search for a comprehensive and evenly distributed set of Pareto optimal solutions.

Table 2. The average and standard deviation of *HV* measures for 5 algorithms

| **Test function** | **Algorithm** | | | | |
|---|---|---|---|---|---|
| | Omni-optimizer | MO-Ring-PSO-SCD | DN-NSGA-II | TriMOEA-TA&R | MPSO |
| MMF1 | 3.67±3.23e-05 | 3.66±4.98e-04 | 3.66±1.54e-03 | 3.66±1.32e-03 | 3.67±3.01e-05 |
| MMF3 | 3.66±6.56e-05 | 3.65±6.12e-03 | 3.65±4.87e-04 | 3.64±1.04e-03 | 3.64±4.73e-03 |
| SYM-PART simple | 1.64±3.58e-04 | 1.62±1.06e-03 | 1.66±2.36e-03 | 1.62±5.11e-03 | 1.61±1.29e-04 |
| SYM-PART rotated | 1.57±3.61e-04 | 1.40±2.93e-03 | 1.58±4.37e-04 | 1.44±1.89e-03 | 1.56±3.73e-04 |
| Omni-test | 62.27±2.16e-04 | 61.13±2.85e-04 | 62.35±3.40e-04 | 62.05±1.57e-04 | 62.17±8.16e-04 |

5. **Conclusions.** Building appropriate mathematical models and applying the MPSO algorithm for structural optimization are important areas of research. This paper explores the significance of using the MPSO algorithm in engineering practice and the need for parameter studies. The article starts by presenting the fundamental principles and application context of the PSO algorithm. It emphasizes the difficulty of achieving a balance between global search and local search, primarily arising from the mapping of the decision space to the objective space. To overcome this, the article draws inspiration from the flower pollination algorithm's global and local pollination transformation probabilities. It proposes a linearly decreasing transformation probability method to enhance the particle's position update strategy, thereby improving the MPSO algorithm's global exploration capability and accuracy. The article also introduces the strategy of linearly decreasing transformation probability and the non-overlapping ring topology structure to develop the MPSO algorithm. It discusses how these techniques impact the population's exploration capability and search accuracy. Experimental findings show that the utilization of a non-overlapping ring topology structure significantly boosts the algorithm's performance. This structure prevents the overwhelming influence of non-dominated solutions within the population, allowing for a more refined and precise search process. These research findings have significant implications for guiding structural optimization in practical engineering. Furthermore, they lay the foundation for exploring the application of optimization algorithms in other domains.

## REFERENCES


[1] A. LaTorre, D. Molina, E. Osaba, J. Poyatos, J. D. Ser, and F. Herrera, "A Prescription Of Methodological Guidelines For Comparing Bio-Inspired Optimization Algorithms," *Swarm and Evolutionary Computation*, vol. 67, 100973, 2021.

[2] L. Gao, D. Gong, N. Cui, M. Lv, and Y. Feng, "Evaluation of Bio-Inspired Optimization Algorithms Hybrid with Artificial Neural Network for Reference Crop Evapotranspiration Estimation," *Computers and Electronics in Agriculture*, vol. 190, 106466, 2021.

[3] F. Wang, H. Zhang, and A. Zhou, "A Particle Swarm Optimization Algorithm for Mixed-Variable Optimization Problems," *Swarm and Evolutionary Computation*, vol. 60, 100808, 2021.

[4] N. Zeng, Z. Wang, W. Liu, H. Zhang, K. Hone, and X. Liu, "A Dynamic Neighborhood-Based Switching Particle Swarm Optimization Algorithm," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 9290-9301, 2022.

[5] X. Fan, W. Sayers, S. Zhang, Z. Han, L. Ren, and H. Chizari, "Review and Classification of Bio-inspired Algorithms and Their Applications," *Journal of Bionic Engineering*, vol. 17, no. 3, pp. 611-631, 2020.

[6] A. P. Piotrowski, J. J. Napiorkowski, and A. E. Piotrowska, "Population size in Particle Swarm Optimization," *Swarm and Evolutionary Computation*, vol. 58, 100718, 2020.

[7] S. Mirjalili, and A. Lewis, "S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization," *Swarm and Evolutionary Computation*, vol. 9, pp. 1-14, 2013.

[8] X. Yang, J. Yuan, J. Yuan, and H. Mao, "A modified particle swarm optimizer with dynamic adaptation," *Applied Mathematics and Computation*, vol. 189, no. 2, pp. 1205-1213, 2007.

[9] N. Jin, and Y. Rahmat-Samii, "Advances in Particle Swarm Optimization for Antenna Designs: Real-Number, Binary, Single-Objective and Multiobjective Implementations," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 556-567, 2007.

[10] X.-F. Liu, Z.-H. Zhan, Y. Gao, J. Zhang, S. Kwong, and J. Zhang, "Coevolutionary Particle Swarm Optimization with Bottleneck Objective Learning Strategy for Many-Objective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 587-602, 2019.

[11] S. Kiranyaz, J. Pulkkinen, and M. Gabbouj, "Multi-dimensional particle swarm optimization in dynamic environments," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2212-2223, 2011.

[12] M. H. Moradi, and M. Abedinie, "A combination of Genetic Algorithm and Particle Swarm Optimization for optimal DG location and sizing in distribution systems," *International Journal of Electrical Power & Energy Systems*, pp. 858-862, 2012.

[13] N. D. Cesare, D. Chamoret, and M. Domaszewski, "A new hybrid PSO algorithm based on a stochastic Markov chain model," *Advances in Engineering Software*, vol. 90, pp. 127-137, 2015.

[14] H. Cao, W. Sun, Y. Chen, F. Kong, and L. Feng, "Sizing and shape optimization of truss employing a hybrid constraint-handling technique and manta ray foraging optimization," *Expert Systems with Applications*, vol. 213, 118999, 2023.

[15] A. Kaveh, and S. Talatahari, "Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures," *Computers & Structures*, vol. 87, no. 5-6, pp. 267-283, 2009.

[16] S. O. Degertekin, L. Lamberti, and I. B. Ugur, "Discrete sizing/layout/topology optimization of truss structures with an advanced Jaya algorithm," *Applied Soft Computing*, vol. 79, pp. 363-390, 2019.

[17] A. Kaveh, and M. I. Ghazaan, "Enhanced colliding bodies algorithm for truss optimization with frequency constraints," *Journal of Computing in Civil Engineering*, vol. 29, no. 6, 04014104, 2015.

[18] C. Dimou, and V. Koumousis, "Reliability-based optimal design of truss structures using particle swarm optimization," *Journal of Computing in Civil Engineering*, vol. 23, no. 2, pp. 100-109, 2009.

[19] F. Pace, A. Santilano, and A. Godio, "A Review of Geophysical Modeling Based on Particle Swarm Optimization," *Surveys in Geophysics*, vol. 42, no. 3, pp. 1-45, 2021.

[20] T.-Y. Wu, H. Li, S. Kumari, and C.-M. Chen, "A Spectral Convolutional Neural Network Model Based on Adaptive Fick's Law for Hyperspectral Image Classification," *Computers, Materials & Continua*, vol. 79, no. 1, pp. 19-46, 2024.

[21] T.-Y. Wu, A. Shao, and J.-S. Pan, "CTOA: Toward a Chaotic-Based Tumbleweed Optimization Algorithm," *Mathematics*, vol. 11, no. 10, 2339, 2023.

[22] T.-Y. Wu, H. Li, and S.-C. Chu, "CPPE: An Improved Phasmatodea Population Evolution Algorithm with Chaotic Maps," *Mathematics*, vol. 11, no. 9, 1977, 2023.

[23] C.-M. Chen, S. Lv, J. Ning, and J. M.-T. Wu, "A Genetic Algorithm for the Waitable Time-Varying Multi-Depot Green Vehicle Routing Problem," *Symmetry*, vol. 15, no. 1, 124, 2023.

[24] G. Sun, D. Qin, T. Lan, and L. Ma, "Research on Clustering Routing Protocol Based on Improved PSO in FANET," *IEEE Sensors Journal*, vol. 21, no. 23, pp. 27168-27185, 2021.

[25] Q. Gu, Q. Wang, X. Li, and X. Li, "A surrogate-assisted multi-objective particle swarm optimization of expensive constrained combinatorial optimization problems," *Knowledge-based Systems*, vol. 223, 107049, 2021.

[26] Z.-Z. Liu, and Y. Wang, "Handling Constrained Multiobjective Optimization Problems with Constraints in Both the Decision and Objective Spaces," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 870-884, 2019.

[27] G. Selvachandran, S. G. Quek, L. T. H. Lan, L. H. Son, N. L. Giang, W. Ding, M. Abdel-Basset, and V. H. C. Albuquerque, "A New Design of Mamdani Complex Fuzzy Inference System for Multiattribute Decision Making Problems," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 4, pp. 716-730, 2020.